ISSN 0181-7450

F : 15,50 €

(B) 16,50 €
(CH) 29,00 FS
(CAN) 20.99 $ca
(And) 15,50 €
DOM surface 16,50 €
DOM avion 16,90 €
N Cal/S 2000 cfp

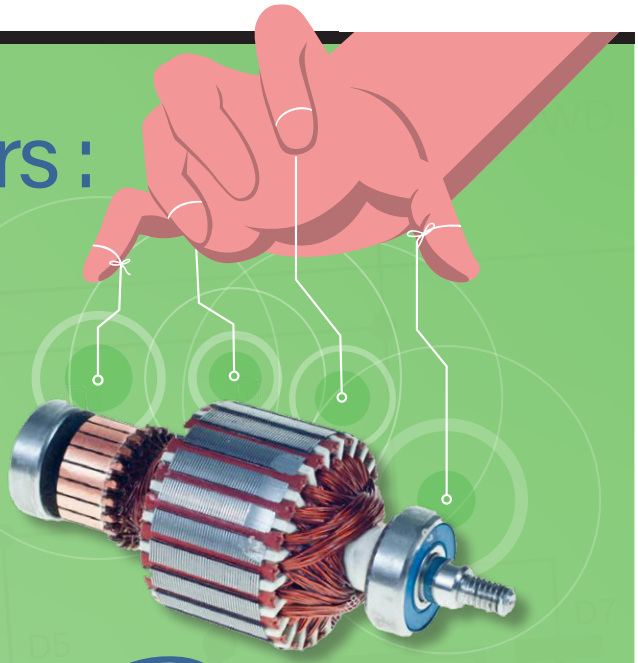
elektor 493

créer > partager > vendre

JANVIER & FÉVRIER 2022
ELEKTORMAGAZINE.FR

Commande de moteurs : les ponts en H

p. 6



FOCUS SUR

Commande des **moteurs**

Testeur de servo polyvalent

p. 73

Moteurs électriques de grande taille

p. 52

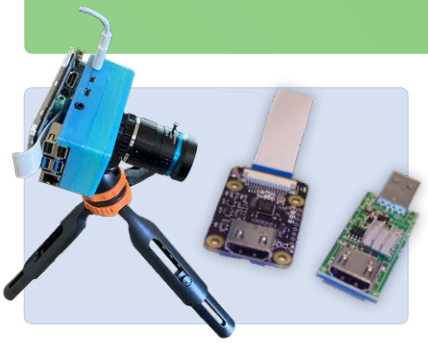
dans ce numéro :

- Interrupteur général pour paillasse (DIY)
- Le Raspberry Pi Zero 2 W passe au quadricœur
- L'équipe d' Elektor Lab
- Carte son au format Raspberry Pi Zero
- Voyage dans les réseaux neuronaux
- Dans l'intimité d'un processeur *open source*

ELEKTOR INDUSTRIE

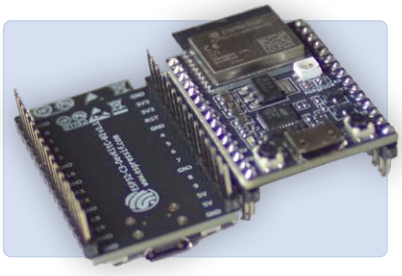
- La commande des moteurs devient simple
- Notes du Forum mondial de l'électronique éthique 2021

et bien d'avantage !



Raspberry Pi comme télécommande KVM
Test du logiciel Pi-KVM

p. 22



Prise en main du microcontrôleur ESP32-C3 RISC-V

p. 59

Véhicule autonome avec lidar 2D Un ESP32 Pico interprète les données du lidar

p. 36

L 19624 - 493 - F: 15,50 € - RD

NOTRE GAMME
PAR DES TECHNICIENS POUR LES TECHNICIENS

The best part of your project: www.reichelt.com/gamme

Uniquement le meilleur pour vous - provenant de plus de 900 marques

Nos responsables produits sont employés par Reichelt depuis de nombreuses années et connaissent les exigences de nos clients. Ils rassemblent une large gamme de produits de qualité, à la fois parfaits pour les besoins dans les domaines de la recherche et du développement, la maintenance, l´infrastructure informatique et la production en petites séries et adaptés pour les fabricants.

Rapport qualité/prix imbattable pour un service Pick and Place professionnel

UFactory xArm5 Lite

Qu'il s'agisse de tâches de positionnement, d'assemblage de composants ou de service au bar entièrement automatisé : avec les bras robotisés de la série xArm, optimisez votre concept d'automatisation, et ce à un coût raisonnable.

- Degré de liberté : 5
- Répétabilité : 0,1 mm
- Portée : 700 mm
- Capacité de charge : 3 kg
- Système : ROS, Python, C++
- Protocole : RS485

UFACTORY

INFOGRAPHIES :

Industrie : Près de la moitié des entreprises industrielles françaises qui utilisent l'IA dans leur production n'ont pas encore de stratégie associée.

En savoir plus
▶ https://**rch.lt/MG726f**

N° de commande :
UF XARM5 LITE

6.600,00 (5.500,00)

Découvrez notre importante gamme robotique ▶
reichelt.com/robotique

Types de paiement : Sofort. VISA MasterCard Maestro PayPal amazon pay

PRIX DU JOUR! Prix à la date du: 15. 12. 2021

- Excellent rapport qualité prix
- Plus de 120 000 produits sélectionnés
- Livraison fiable - depuis l'Allemagne dans le monde entier

www.reichelt.com

Assistance téléphonique: +33 97 518 03 04

reichelt elektronik – Tirer le meilleur parti de votre projet

Les réglementations légales en matière de résiliation sont applicables. Tous les prix sont indiqués en € TVA légale incluse, frais d'envoi pour l'ensemble du panier en sus. Seules nos CGV sont applicables (sur le site https://rch.lt./CG-FR ou sur demande). Semblables aux illustrations. Sous réserve de coquilles, d'erreurs et de modifications de prix. reichelt elektronik GmbH & Co. KG, Elektronikring 1, 26452 Sande (Allemagne), tél. +33 97 518 03 04

45ème année
n° 493 – janvier-février 2022

ISSN 0181-7450
Dépôt légal : janvier 2022
CPPAP 1125 T 83713
Directeur de la publication : Donatus Akkermans

Elektor est édité par :
PUBLITRONIC SARL
c/o Regus Roissy CDG
1, rue de la Haye
BP 12910
FR - 95731 Roissy CDG Cedex

Pour toutes vos questions :
service@elektor.fr

www.elektor.fr | www.elektormagazine.fr

Banque ABN AMRO : Paris
IBAN : FR76 1873 9000 0100 2007 9702 603
BIC : ABNAFRPP

Publicité :
Raoul Morreau
Tél. : +31 (0)6 4403 9907
Courriel : raoul.morreau@elektor.com

DROITS D'AUTEUR :
© 2022 Elektor International Media B.V.

Toute reproduction ou représentation intégrale ou partielle, par quelque procédé que ce soit, des pages publiées dans la présente publication, faite sans l'autorisation de l'éditeur est illicite et constitue une contrefaçon. Seules sont autorisées, d'une part, les reproductions strictement réservées à l'usage privé du copiste et non destinées à une utilisation collective, et, d'autre part, les analyses et courtes citations justifiées par le caractère scientifique ou d'information de l'oeuvre dans laquelle elles sont incorporées (Loi du 11 mars 1957 -art. 40 et 41 et Code Pénal art. 425).
Certains circuits, dispositifs, composants, etc. décrits dans cette revue peuvent bénéficier de droits propres aux brevets; la Société éditrice n'accepte aucune responsabilité du fait de l'absence de mention à ce sujet. Conformément à l'art. 30 de la Loi sur les Brevets, les circuits et schémas publiés dans Elektor ne peuvent être réalisés que dans des buts privés ou scientifiques et non commerciaux. L'utilisation des schémas n'implique aucune responsabilité de la part de la Société éditrice. La Société éditrice n'est pas tenue de renvoyer des articles qui lui parviennent sans demande de sa part et qu'elle n'accepte pas pour publication. Si la Société éditrice accepte pour publication un article qui lui est envoyé, elle est en droit de l'amender et/ou de le faire amender à ses frais; la Société éditrice est de même en droit de traduire et/ou de faire traduire un article et de l'utiliser pour ses autres éditions et activités, contre la rémunération en usage chez elle.

Imprimé aux Pays-Bas par Pijper Media – Groningen
Distribué en France par M.L.P. et en Belgique par A.M.P.

rédacteur en chef d'Elektor Magazine

Commande des moteurs et plus encore

Grâce à des études de notre lectorat, nous savons que la plupart des lecteurs d'*Elektor* sont des électroniciens professionnels. Ils travaillent dans des bureaux d'études, dans des services d'achat ou de vente de composants et d'appareils... Nous avons donc toujours été un magazine qui sourit de la distinction entre électronique de loisir et électronique professionnelle. Après tout, plus d'un concepteur professionnel a tiré une inspiration précieuse de nos projets présentés de manière ludique. Et à l'heure où les makers fondent leurs propres entreprises ou sont courtisés par les départements de R&D des grandes entreprises de semi-conducteurs, cette façon de voir en noir et blanc est probablement définitivement obsolète.
Il y a quelques années, nous avons lancé notre magazine « spécial industrie » (d'abord sous le nom de « Elektor Business »). Vous pouvez y découvrir en avant-première ce que les départements de recherche et développement des fabricants s'apprêtent à mettre sur le marché. Vous y trouvez également de passionnants articles de synthèse sur les dernières tendances du secteur (rédigés par exemple par Stuart Cording). Cet ingénieur se déplace pour nous sur les salons professionnels et entretient d'excellentes relations avec de nombreux experts qui lui fournissent des informations de fond.
En 2022, nous reviendrons aux numéros thématiques, à savoir l'exploration d'un sujet en particulier, et nous intégrerons les articles « spécial industrie » les plus intéressants. En nous concentrant davantage sur un domaine – sans sacrifier la couverture de l'ensemble du spectre de l'électronique dans nos numéros – nous pourrons examiner différentes disciplines de manière plus complète et approfondie.
Dans ce numéro, nous nous intéressons tout particulièrement à la « commande des moteurs ». Aux pages 6, 52 et 73, vous trouverez deux articles de fond de Mathias Claußen et Thomas Scherer, ainsi que (bien sûr !) un projet de circuit sur le sujet. Et dans les pages « industrie », Stuart Cording explique comment simplifier la commande de moteurs.

Bon début d'année 2022 !
Restez curieux !

notre équipe

Rédacteur en chef : Jens Nickel
Rédaction : Eric Bogers, Jan Buiting, Rolf Gerstendorf, Thomas Scherer, Clemens Valens, Mariline Thiebaut-Brodier (coordination)
Service aux lecteurs : Ralf Schmiedel
Correcteur technique : Malte Fischer
Laboratoire : Mathias Claussen, Ton Giesberts, Luc Lemmens, Clemens Valens (responsable)
Maquette : Giel Dols, Harmen Heida

Elektor est membre de la FIPP, une organisation qui « se développe depuis presque 100 ans pour réunir des propriétaires de médias et des créateurs de contenu du monde entier ».

Elektor est membre de VDZ (association d'éditeurs de magazines allemands) qui « représente les intérêts communs de 500 éditeurs allemands grand public et B2B. »

DANS CE NUMÉRO

Rubriques

3 **Édito**

72 **Gagnants du concours Fast Forward 2021**
du salon productronica

92 **Démarrer en électronique... (11)**
Toujours la bobine, mais avec ses applications pratiques dans l'audio.

96 **Projet 2.0**
Corrections, mises à jour et courriers des lecteurs

110 **Questions d'éthique**
Pauvreté et électronique

114 **Hexadoku**
Casse-tête pour elektorniciens

Articles de fond

FOCUS
6 **Commande de moteurs : les ponts en H**
Introduction aux moteurs CC, pas à pas et sans balais

16 **L'équipe d' Elektor Lab**
Notre approche, nos outils préférés et plus encore

28 **IQaudio Codec Zero**
Carte son au format Raspberry Pi Zero

31 **Le projet PiKVM et ses enseignements**
Entretien avec Maxim Devaev

40 **Le Raspberry Pi Zero 2 W passe au quadricœur**

FOCUS
52 **Moteurs électriques de grande taille**
Principes de base et informations utiles

68 **Création d'interfaces graphiques en Python**
2e partie : mon nom n'est pas Bond

82 **Voyage dans les réseaux neuronaux**
3e partie : les neurones pratiques

87 **Dans l'intimité d'un processeur *open source***
Extrait : résultats des FPGA Lattice et Xilinx

102 **Sourcemètre BattLab-One**
Mesurer et optimiser la durée de vie des batteries des appareils IdO

Industrie

44 **Notes du Forum mondial de l'électronique éthique 2021**

FOCUS
46 **Commande de moteur**
La commande des moteurs devient simple

L'équipe d'elektor LAB

Notre approche, nos outils préférés et plus encore

16

le Raspberry Pi Zero 2 W **passe au quadricœur**

40

protégez-vous et protégez les autres !

Interrupteur général pour paillasse (DIY)

64

Réalisations

22 **Raspberry Pi comme télécommande KVM**
Test du logiciel Pi-KVM

36 **Véhicule autonome avec lidar 2D**
Un ESP32 Pico interprète les données du lidar

59 **Prise en main du microcontrôleur ESP32-C3 RISC-V**

64 **Protégez-vous et protégez les autres !**
Interrupteur général pour paillasse (DIY)

FOCUS
73 **testeur de servo polyvalent**
Vérification du comportement en l'absence de fiche technique

78 **Modbus sans fil (partie 2)**
Logiciel de la carte Modbus TCP sans fil

98 **De la couleur au son**
Comment exploiter un capteur de couleurs avec l'I²C

106 **Traceur de fuites à la terre simple**
Mesure de l'isolation du secteur

Traceur de fuites à la terre simple
Mesure de l'isolation du secteur

106

Bientôt dans ces pages

Le numéro de mars-avril 2022 d'Elektor
Vous retrouverez dans le prochain magazine Elektor l'habituel mélange stimulant de réalisations originales, de circuits soigneusement étudiés, d'articles de fond, de sujets nouveaux, de trucs et d'astuces pour les électroniciens actifs. Le thème principal sera « l'électronique embarquée ».

Quelques-uns des points forts :

- Atelier : cœur RISC-V sur FPGA
- Carte tampon pour le Raspberry Pi 400
- Contrôleurs à 4 bits et autres contrôleurs à bas prix
- Collection de printemps de cartes Raspberry Pi RP2040
- Instrument de mesure portatif pour la température et l'humidité
- Mesures audio avec interface audio USB
- WinUI 3 : nouveau cadriciel pour les applications Windows
- Mises à jour en temps réel pour Arduino et ESP
- NB-IoT en pratique

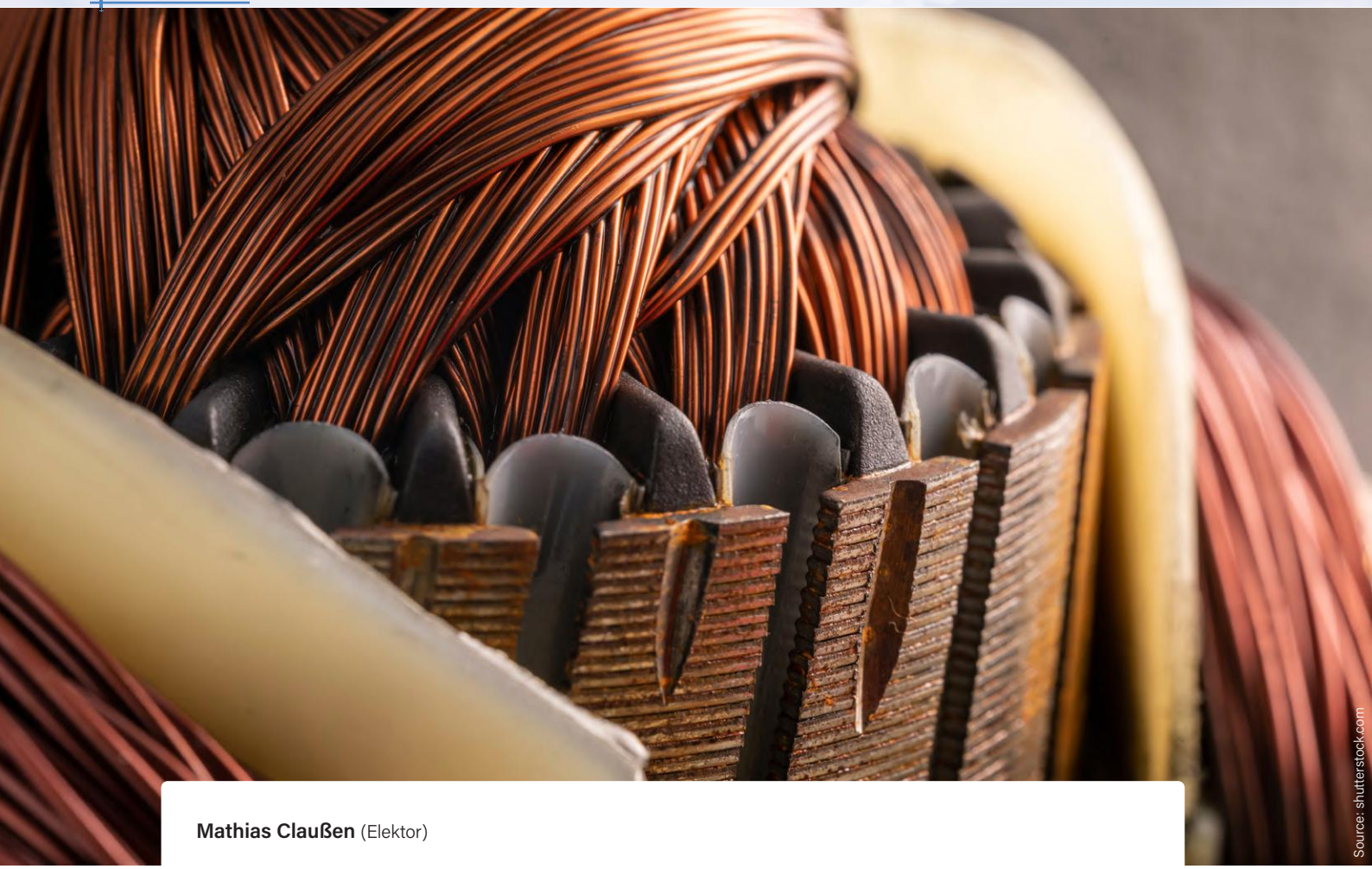
et bien d'autres choses encore !

Le numéro de mars-avril 2022 du magazine Elektor sera publié aux alentours du 10 mars 2022. La date d'arrivée du magazine papier chez les abonnés dépend des aléas d'acheminement. Le contenu et les titres des articles peuvent être modifiés.

commande de moteurs : les ponts en H

Introduction aux moteurs CC, pas à pas et sans balais

Motor Control



Source: shutterstock.com

Mathias Claußen (Elektor)

Commander des moteurs est nécessaire dans bien d'autres domaines que la ventilation. Les moteurs des convoyeurs, des pompes ou des imprimantes 3D ne peuvent se contenter d'un mode marche-arrêt. Il faut aussi commander la puissance et le sens de rotation. Pour rester accessible, cette introduction n'abordera que les moteurs CC classiques, pas à pas et *brushless* (sans balais) de faible puissance.

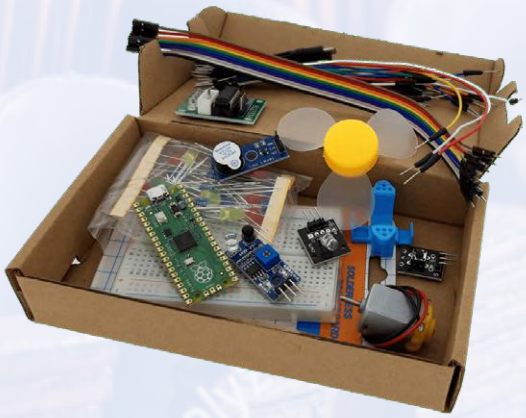


Figure 1. Le kit de base pour Raspberry Pi Pico avec un moteur CC. (Source : Kuongshun)

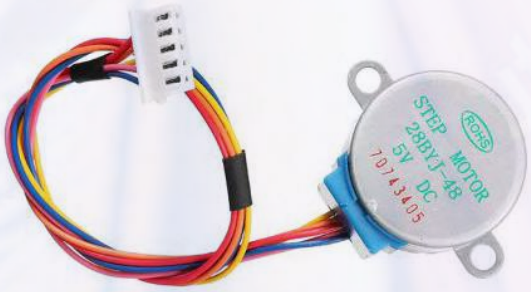


Figure 2. Moteur pas à pas de type 28BYJ-48 pour 5 V CC. (Source : Geekcreit)

Les circuits présentés dans cet article ont pour but d'illustrer nos propos. Ce sont des points de départ pour vos propres réalisations et non des circuits à copier tels quels. Voyons d'abord un moteur CC simple. Il s'agit ici d'un moteur de 3 à 6 V de tension de service et 1 A de courant de service maximal. On trouve des moteurs de ce type dans de nombreux kits de base, par ex. celui du Raspberry Pi Pico [1] (**fig. 1**). Ce kit s'alimente en 5 V, tension fournie par un port USB standard. Certains kits de démarrage incluent aussi un moteur pas à pas ou le proposent contre un supplément modique. Le 28BYJ-48 [2] (**fig. 2**), qui peut être alimenté en 5 V, est un type courant.

En avant et en arrière

Examinons le circuit de la **figure 3**. Le relais K1 agit comme interrupteur M/A pour le moteur. Le relais est commandé par le contact à action momentanée S1, ainsi, le courant requis par le moteur ne traverse que le relais, et non le contact. La diode D1 supprime le pic de tension aux bornes de la bobine du relais à l'ouverture du contact.

En actionnant le contact à action momentanée S1, le relais K1 colle et le moteur M1 commence à tourner. Lorsque le contact S1 est relâché, le relais retombe et le moteur n'est plus alimenté. Comme les bobines des relais, les enroulements des moteurs CC peuvent produire des pics de tension. La diode D2 agit comme la diode D1 ci-dessus et évite que le moteur ne produise une pointe de tension indésirable à la coupure de K1. Ce circuit ne permet que de faire tourner le moteur dans un seul sens. Mais que faire si le moteur doit aussi tourner en arrière (par ex. pour actionner un treuil) ? Il faut modifier le circuit pour que cela soit possible.

Le *pont en H* est une bonne solution. Voir l'exemple de la **figure 4**. Le nom est basé sur la structure du schéma du circuit. Le moteur est connecté au milieu du H. En actionnant le contact S1 (marche avant), les relais K1 et K4 collent et le courant traverse le moteur de sa borne positive à sa borne négative. Pour que le moteur tourne en arrière, il suffit d'actionner le contact S2 et les relais K3 et K2 collent. Le courant traverse alors le moteur de la borne négative à la borne positive, et il tourne en marche arrière. Noter qu'il ne faut jamais actionner les deux contacts à action momentanée (S1 et S2) à la fois. Si les relais K1 et K2, ou K3 et K4, collaient en même temps, leurs contacts court-circuiteraient l'alimentation avec un courant élevé qui pourrait les endommager gravement.

De la lenteur à la rapidité

Avec le pont en H, nous pouvons maintenant commander le sens de rotation du moteur. Ce n'est toutefois pas le cas de la vitesse du moteur. La commutation des relais sous charge, surtout en CC, réduit notablement leur durée de vie. Au-delà de cette usure rapide, les relais sont encombrants et souffrent d'une faible efficacité énergétique.

La première amélioration logique est l'emploi de semi-conducteurs. Voyons un schéma avec des transistors NPN et PNP. La **figure 5** donne le schéma du circuit modifié. Les relais ont été remplacés par une paire de transistors PNP (T1 et T3) et une paire de transistors NPN (T2 et T4). Le circuit de commande a également été modifié. Maintenant, le contact S1, s'il est fermé, relie les bases de T1 et T2 à la terre à travers les résistances de limitation de courant R1 et R2, ou s'il est ouvert, relie ces bases à la tension d'alimentation à travers R5. Le contact S2 agit de même sur les bases de T3 et T4.

Au repos du circuit avec S1 et S2 ouverts, la tension d'alimentation est appliquée aux résistances R1 à R4 à travers les résistances R5 et R6. Par conséquent les transistors T1 et T3 sont coupés et ont donc une impédance élevée (de l'ordre du mégohm). En revanche, les transistors T2 et T4 sont mis en conduction par leur courant de base et ont donc une faible impédance. C'est pourquoi les deux bornes du moteur sont reliées à la masse et aucun courant n'y circule.

Que se passe-t-il maintenant si le contact S1 est fermé et S2 ouvert ? Dans ce cas, les résistances R1 et R2 sont reliées à la masse. La base et l'émetteur du transistor T2 sont au même potentiel et le courant de base cesse. Le transistor T1, en revanche, est conducteur car le courant peut circuler de l'émetteur à la base. Par conséquent, le positif de l'alimentation est appliqué à une borne du moteur tandis que l'autre est reliée à la masse : le moteur commence à tourner. Si le contact S1 est ouvert et l'interrupteur S2 fermé, alors le transistor T3 est passant et le transistor T4 coupé. Dans ce cas, le moteur commence aussi à tourner, mais dans l'autre sens.

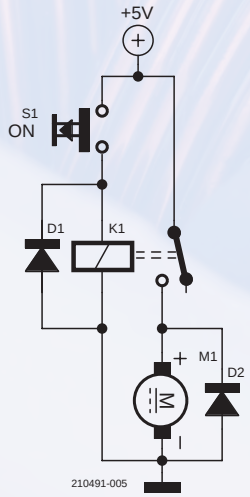


Figure 3. Commande de moteur par des relais.

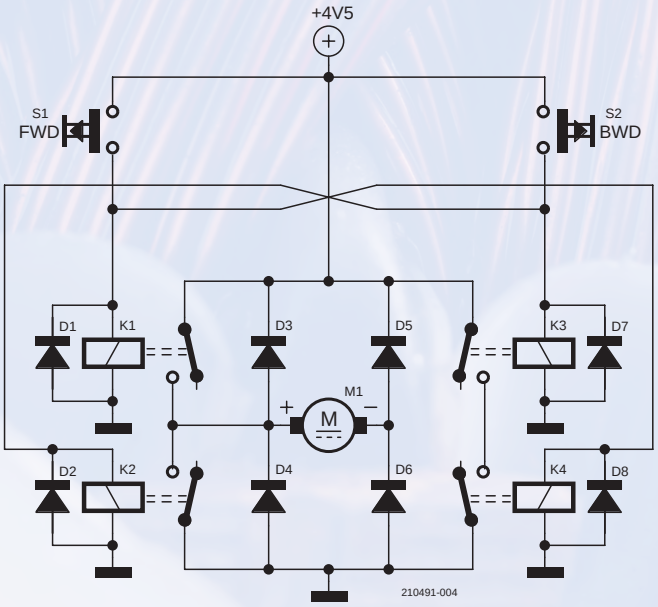


Figure 4. Pont en H avec relais.

Comment faire alors pour commander la vitesse du moteur, par ex. à l'aide d'un microcontrôleur (µC) ? Deux broches de sortie d'un µC joueront le rôle des contacts. Elles sont notées *PWM_A* et *PWM_B* sur la **figure 6**. Pour que le moteur avance ou recule, il suffit de piloter l'état de ces broches de façon adéquate. La *modulation de largeur d'impulsion* (MLI, ou PWM en anglais) peut contrôler la vitesse. Celle-ci est alors déterminée par le rapport cyclique (temps d'activation/période) du signal MLI, et la direction est dictée par la broche de sortie de l'onde impulsionnelle *PWM_A* ou *PWM_B*. Ce schéma est conçu pour être alimenté en 5 V, les sorties du µC doivent donc pouvoir gérer ce niveau de signal. Si le moteur doit s'alimenter sous une tension plus élevée ou si le µC ne peut gérer que le 3,3 V sur ses broches d'E/S (par ex. Raspberry Pi), le circuit doit être légèrement modifié.

La **figure 7** répond au problème avec deux transistors supplémentaires (T5 et T6) pilotés par les signaux MLI à travers des résistances de base. Ces transistors permettent d'utiliser des µC fonctionnant à une tension d'alimentation inférieure à celle du moteur CC. Il faut noter ici que les signaux MLI sont maintenant inversés. Une tension appliquée au transistor T5 ou T6, le met en conduction en tirant à la masse les entrées des transistors T2 et T1, ou T3 et T4. Il faut en tenir compte dans les signaux de commande délivrés par le µC, car cela inverse les sens avant et arrière.

Malgré tout, l'utilisation de transistors PNP et NPN présente des inconvénients. Le courant de base n'est pas négligeable, il y a donc perte d'énergie, et la fréquence du signal MLI est limitée. Le passage de l'impédance de l'état haut à l'état bas n'est pas instantané, il y a donc des pertes de commutation. En outre, au moment où T1 et T2, ou bien T3 et T4, changent d'état, il y a un bref intervalle durant lequel les deux transistors de la paire sont conducteurs. Ainsi un courant plus élevé peut circuler directement de la tension d'alimentation à la masse. Cela augmente aussi la perte de puissance et peut même causer une défaillance précoce.

Ponts en H avec FET

Le choix de transistors à effet de champ (FET) permet d'éviter certains

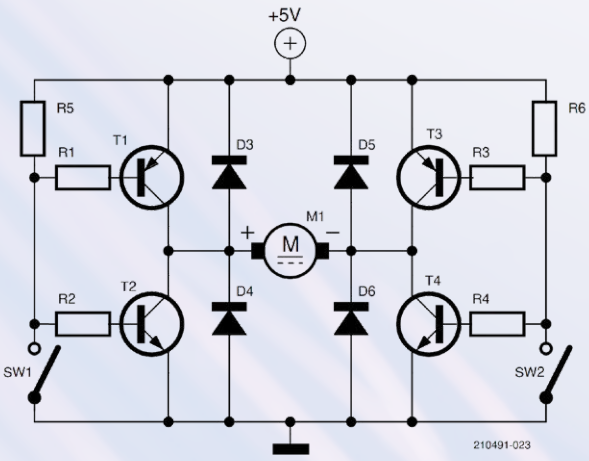


Figure 5. Pont en H avec transistors NPN et PNP.

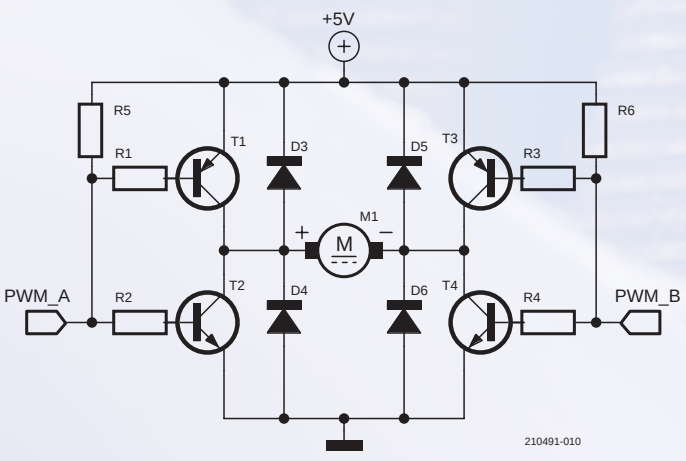


Figure 6. Commande par signaux MLI provenant µC.

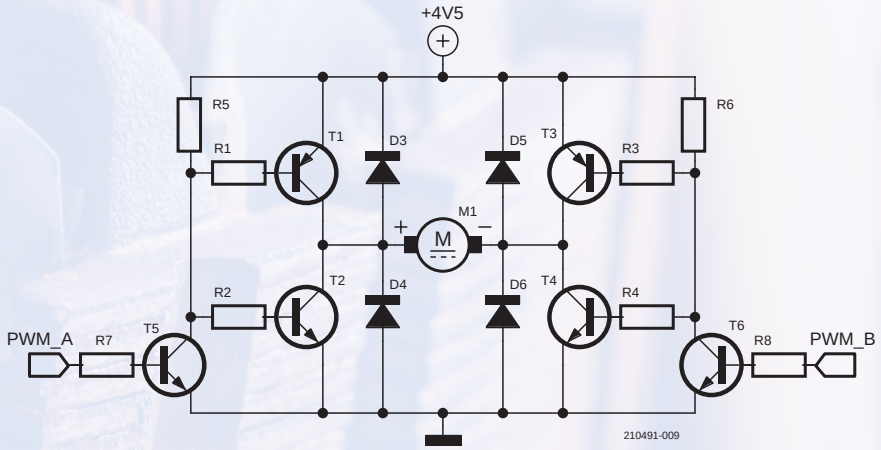


Figure 7. Commande par signaux de 3,3 V.

des inconvénients des transistors bipolaires. Ils peuvent repousser les limites : vers le haut en fréquence et vers le bas en perte de puissance. Tous les ponts en H évoqués ci-dessus ont ce problème de court-circuit transitoire lors de la commutation des transistors. Cet écueil doit être éliminé. Le pont en H de la **figure 8** est donc construit avec des FET à commande individuelle.

T1 et T3 sont des FET PMOS, tandis que T2 et T4 sont des FET NMOS. Un FET PMOS est coupé (haute impédance) lorsque la différence de tension entre la grille et la source est proche de zéro. Pour mettre en conduction (basse impédance) un FET PMOS, la tension de grille doit être plus négative que la tension de source. Pour T1 et T3, cela signifie que lorsque la grille est tirée à la masse, le FET est conducteur. La tension grille/source maximale admissible doit également être prise en compte. Si cette tension est dépassée, le FET sera endommagé. Pour mettre en conduction un FET NMOS, la tension de grille doit être plus positive que la tension de source. Pour T2 et T4, cela signifie qu'ils sont dans un état de haute impédance lorsque la grille est tirée à la masse, et qu'ils commencent à conduire si la grille est à une tension positive (ici par rapport à la masse).

Sur la figure 8, les diodes D4 à D6 sont toujours là, mais elles ne sont pas nécessaires si des FET sont utilisés car des diodes internes remplissent la même fonction. Le symbole FET de T1 à T4 de la **figure 9** montre clairement la diode interne comme composant de sa structure. Par conséquent, les diodes D4 à D6 peuvent être omises pour aboutir au schéma de la **figure 10**.

Si seule une alimentation de 5 V est disponible, ne peut-on pas simplement relier PWM_A à !PWM_A et PWM_B à !PWM_B ? Oui en théorie, mais cela conduirait à des courants transitoires de court-circuit encore pires qu'avec des transistors bipolaires. À titre d'expérience mentale, supposons que les PWM_A et !PWM_A soient reliés. Avec 5 V appliqués à T1 et T2 comme condition initiale, T1 est conducteur et T2 est coupé. Maintenant, tirons PWM_A et !PWM_A à la masse. Le signal de 5 V tombe à 0 V, mais ce n'est pas instantané. Au contraire, cela

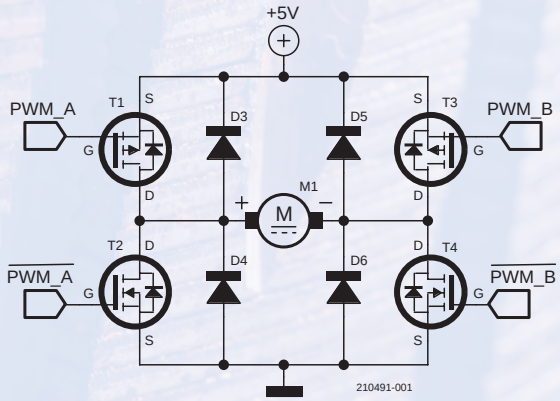


Figure 8. Pont en H avec FET.

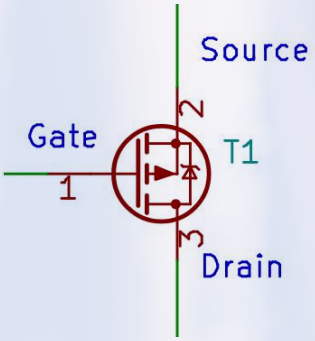


Figure 9. Symbole d'un FET avec diode de suppression intégrée.

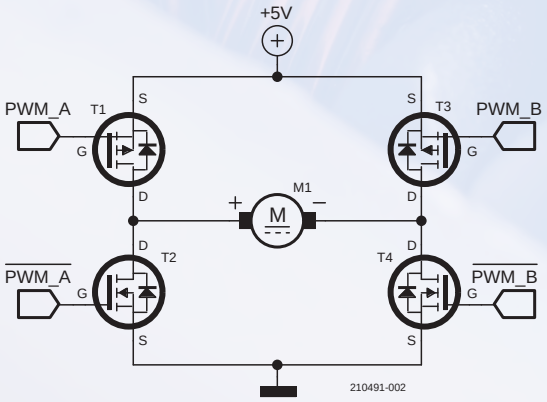


Figure 10 Schéma de circuit sans diodes de suppression.

prend un temps de l'ordre de la nanoseconde. Il y a donc un instant t où les deux FET sont conducteurs : T2 n'est pas encore tout à fait coupé alors que T1 est déjà passant. Dans cette situation, les deux FET n'opposent que qq Ω au courant de court-circuit qui circule directement dans T1 et T2. Cela s'appelle le *shot-through* (tir traversant), et c'est proscrit dans un pont en H car les composants risquent d'être sévèrement endommagés. Il est donc vital de ne pas essayer de connecter les deux signaux. Certaines unités MLI de µC perfectionnées peuvent produire des signaux de commande de ponts en H en insérant des temps morts appropriés pour éviter ces conditions destructives.

Circuit intégré de commande de pont en H

Il faut soit que le µC gère des temps morts garantissant que T1 et T2 ne sont jamais conducteurs en même temps, soit utiliser un circuit intégré (CI) de commande approprié. Le pont en H à transistors FET (deux NMOS et deux PMOS) décrit ci-dessus a un autre inconvénient : la conductivité des FET PMOS est au moins dix fois moins bonne que celle des NMOS. Cela signifie une perte thermique plus forte, surtout avec des courants relativement élevés. La **figure 11** montre un circuit avec deux CI de commande en demi-pont (NCP5901B de ON Semiconductor) et donc câblé avec quatre FET NMOS. Les condensateurs de ce schéma ne sont pas des condensateurs de blocage ; C1 et C2 font partie de la pompe de charge du NCP5901B. Mais pourquoi le CI d'attaque en demi-pont a-t-il besoin d'une pompe de charge ? Ici, T1 et T3 sont des FET NMOS qui, comme vu ci-dessus, ont besoin d'une tension de grille supérieure à celle de source pour devenir passants. Donc pour T1 et T3, cela signifie que la tension de grille doit être supérieure à la tension d'alimentation du pont en H. La pompe de charge permet d'y parvenir. La **figure 12** donne le schéma fonctionnel interne du CI de commande du demi-pont. Le bloc noté *Anti-Cross Conduction* (anticroisement de conduction) sert à produire les temps morts destinés à empêcher le *shot-through*.

Ainsi, avec des FET adaptés à ce cas d'utilisation, on peut construire un pont en H à partir de composants discrets. De tels ponts en H permettent aussi de traiter des courants supérieurs à 25 A. Une solution plus compacte, entièrement intégrée, est possible pour de plus faibles courants.

Le STM L298N (**fig. 13**) abrite deux ponts en H complets et équipe bien des cartes de commande de moteur bon marché. Ce CI n'intègre pas des FET, mais des transistors bipolaires. Il peut fournir jusqu'à 4 A sous 46 V. Un coup d'œil au schéma fonctionnel (**fig. 14**) révèle une structure très familière. Il ne fait pas de doute que les µC à E/S en 3,3 V peuvent piloter le L298N. Ce module est disponible en vente à distance pour environ 3 à 6 €. Le module IFX9201SG d'Infineon est nettement plus compact. Il peut gérer jusqu'à 6 A et dispose de circuits de protection et de diagnostic intégrés, comme le montre la **figure 15**. Il coûte environ 4 € par appareil. Il est facile de concevoir une commande de moteur CC avec ce CI, mais il est indisponible jusqu'à la mi-2022, comme tant d'autres composants à base de semi-conducteurs. S'il vous faut un pont en H prêt à l'emploi pour commander un moteur, jetez un coup d'œil aux cartes Cytron [3] dans la boutique Elektor.

Moteurs pas à pas, étape par étape

Les moteurs pas à pas tournent, mais comme leur nom le suggère, par fractions de rotation appelées pas ou incréments. Ils ne sont pas conçus

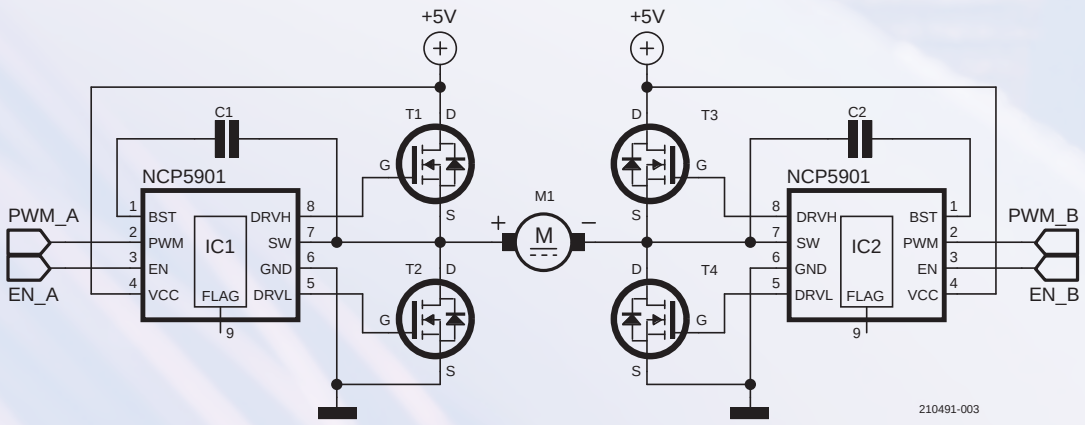


Figure 11. Quatre FET NMOS avec CI de commande.

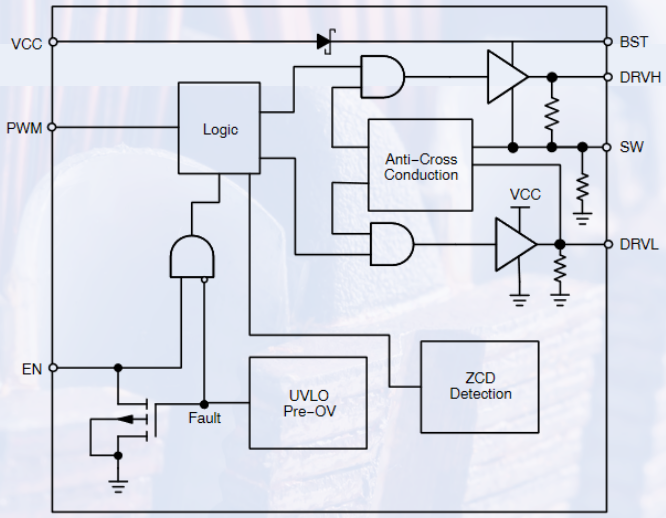


Figure 12. Schéma-bloc interne du NCP5901B de ON Semiconductor (source : ON Semiconductor).

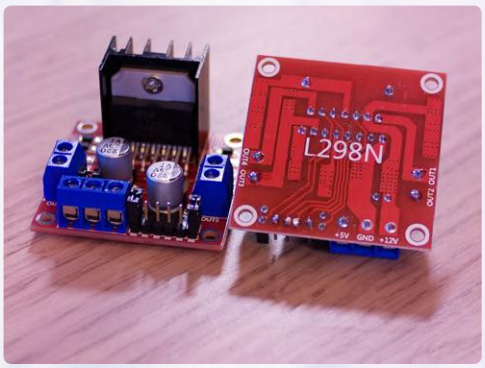


Figure 13. Le module L298N en double pont en H.

pour le fonctionnement continu à vitesse élevée, mais ils peuvent effectuer des rotations incrémentielles à couple élevé. Les moteurs pas à pas sont présents dans la plupart des foyers… Dans les imprimantes à jet d'encre, ils servent à déplacer la tête d'impression ou à l'avance du papier. Un moteur pas à pas fonctionne avec une taille de pas définie (l'incrément), il est donc très facile de calculer la position exacte de la tête d'impression. Certains d'entre nous se souviennent du son des moteurs pas à pas des lecteurs de disquettes, où ils déplaçaient la tête de lecture/écriture.

Mais comment piloter un moteur pas à pas, et pourquoi se déplace-t-il par pas ? Il existe différentes formes de moteurs pas à pas : moteurs pas à pas à réluctance ou à aimant permanent ou encore hybrides. Nous considérons ici les moteurs pas à pas hybrides. Ils se sont imposés dans de nombreuses applications en raison de leur couple élevé et de leur grand nombre de pas par rotation.

La **figure 16** donne la structure de ce type de moteur pas à pas. Un rotor, composé d'aimants permanents et d'anneaux dentés en fer doux prend place à l'intérieur. Le stator est constitué d'un jeu d'enroulements disposés de telle sorte que le rotor ne se déplace que d'un pas à la fois. Côté entraînement, les moteurs pas à pas les plus courants ont deux pôles. D'autres nombres de pôles sont possibles, mais ne seront pas considérés ici. Côté structure, il existe deux types de moteurs pas à pas : unipolaire et bipolaire. Nous allons les examiner ci-dessous.

Moteurs pas à pas unipolaires

Notre moteur pas à pas type sera le 28BYJ-48 illustré à la **figure 2** et présent dans de nombreux kits de démarrage. Ce petit moteur pas à pas unipolaire possède en outre un réducteur d'un rapport de 63,68395:1. Avec un pas de 11,25°, le nombre de pas pour une rotation complète de l'arbre de sortie du réducteur est de 2037,8864, soit environ 2038 pas par tour (arrondi au chiffre supérieur). La connexion du moteur nécessite cinq fils : A, B, C, D, et l'alimentation.

La commande d'un moteur pas à pas unipolaire est très facile, et pour cette taille de moteur, l'ULN2003, disponible auprès de divers fabricants, convient. Comme le montre le schéma-bloc de la **figure 17**, l'ULN2003 renferme sept transistors Darlington. Ce CI comprend des diodes de

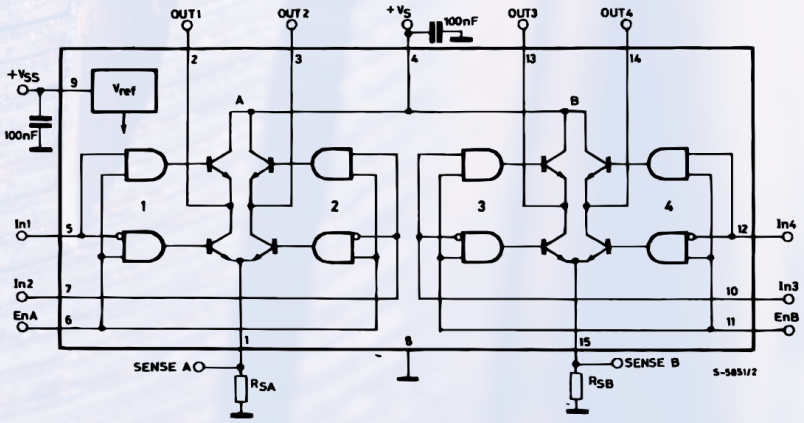


Figure 14. Structure interne du L294N (source : STMicroelectronics).

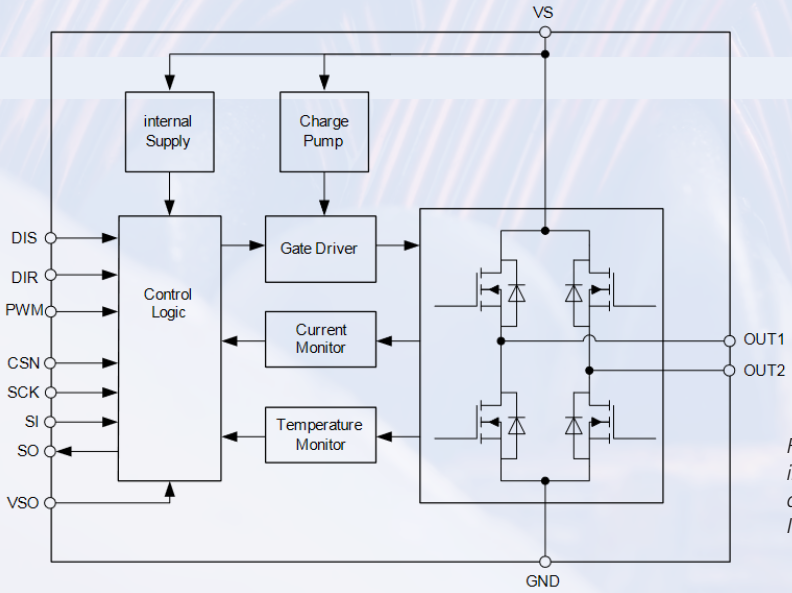


Figure 15. Schéma-bloc interne de l'IFX9201SG d'Infineon (source : Infineon).

suppression, il peut donc tout à fait piloter des charges inductives. La **figure 18** montre un circuit avec ULN2003 utilisable pour piloter un moteur pas à pas 28BYJ-48. Outre le mode pas à pas, le 28BYJ-48 accepte le mode demi-pas. Selon la fiche technique, c'est même la méthode d'entraînement recommandée. Le **tableau 1** donne la séquence des signaux d'entraînement pour une rotation dans le sens horaire. Pour faire un pas dans le sens inverse, il faut parcourir la séquence en sens inverse. Les moteurs pas à pas unipolaires sont devenus rares aujourd'hui car ils ne délivrent que la moitié environ du couple d'un moteur pas à pas bipolaire de même taille. Cependant, ceci est compensé par la commande très simple avec un ULN2003. Il est possible de modifier un moteur pas à pas 28BYJ-48 pour le rendre bipolaire. À cet effet, lisez l'article du blog de jangeox [4]. Ce moteur pas à pas pourra produire plus de couple, mais aura besoin d'un pilote différent.

Moteurs pas à pas bipolaires

Les moteurs pas à pas bipolaires de la plupart des imprimantes 3D n'ont que quatre fils. Pour ces moteurs, il existe des tailles normalisées définies par la *National Electrical Manufactures Association* (NEMA). La NEMA 17-03, par exemple, désigne la taille d'un type de moteur pas à pas produit et/ou vendu par diverses entreprises. Le moteur pas à pas NEMA17-03 de Joy-IT [5] en est un exemple (**fig. 19**). Un ULN2003 ne suffit pas à piloter ce type de moteur pas à pas ; le circuit de commande nécessite deux ponts en H. Un L298N, comme vu ci-dessus pour les moteurs CC, fournit deux ponts en H en un seul boîtier. Il commande le moteur dans le circuit de la **figure 20**. Ces ponts en H étant à transistors bipolaires, des diodes de suppression sont nécessaires aux sorties. La séquence du **tableau 2** doit être exécutée pour obtenir un pas dans le sens horaire.

Au contraire des moteurs unipolaires, l'enroulement complet est utilisé, et non la moitié seulement avec prise centrale. Le couple est plus élevé qu'avec un moteur pas à pas unipolaire de même taille. La commande par ponts en H est très facile, et des CI hautement intégrés comme le MP6500 de MPS ou le TMC2300-LA de Trinamic la rendent encore plus aisée. Ils contiennent une partie de la logique qui exécute les séquences de pas. Le µC ne spécifie que la direction souhaitée (sens horaire ou antihoraire) et fournit une impulsion pour chaque pas souhaité. Les moteurs pas à pas ont l'avantage d'un couple élevé et, à l'inverse des moteurs CC, n'ont pas besoin de balais ni de commutateurs pour acheminer le courant au rotor.

Tableau 1. Séquence d'une étape

Étape/ couleur	Orange	Jaune	Rose	Blue
1.	1	0	0	0
2.	1	1	0	0
3.	0	1	0	0
4.	0	1	1	0
5.	0	0	1	0
6.	0	0	1	1
7.	0	0	0	1
8.	1	0	0	1

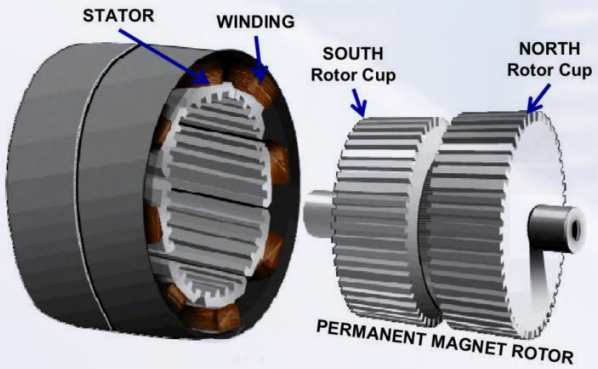


Figure 16. Structure d'un moteur pas à pas (source : Microchip)

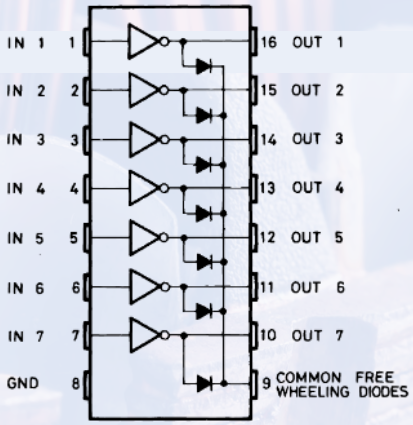


Figure 17. Structure interne de la matrice Darlington ULN2003 (source : STMicroelectronics).

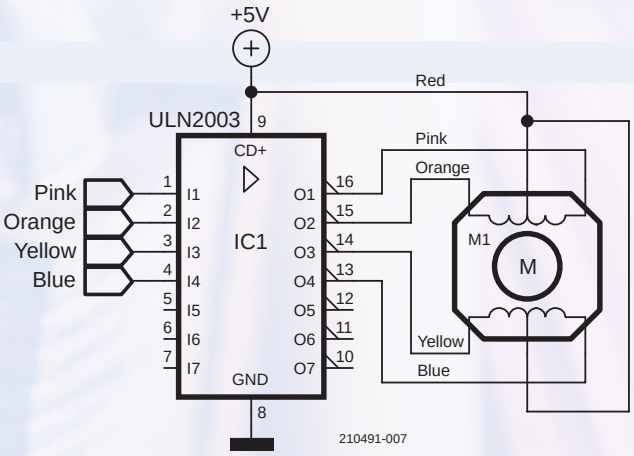


Figure 18. Schéma du circuit avec ULN2003 et moteur pas à pas.

Moteurs CC sans balais

En combinant les propriétés d'un moteur CC avec le couple et la faible maintenance d'un moteur pas à pas, on obtient un moteur CC sans balais. Ces moteurs dits *BLDC* (en anglais *brushless DC* = CC sans balais) sont parfaits pour les tâches d'entraînement dans toutes sortes d'applications. Dans les moteurs CC classiques, les balais conduisent le courant aux enroulements du rotor. À chaque tour, ces balais s'usent et répandent de la poussière conductrice à l'intérieur du moteur, il faut les remplacer à intervalles réguliers et les moteurs peuvent nécessiter des nettoyages. Les moteurs BLDC n'ont pas besoin de cet entretien et ils fournissent un couple élevé dans un petit boîtier. Cependant, la commande des moteurs BLDC est complexe et tout défaut dommageable. C'est pourquoi nous ne donnons ici qu'un bref aperçu de la théorie de la commande des moteurs BLDC.

La structure d'un moteur BLDC est très similaire à celle d'un moteur pas à pas. Il comporte un stator (la partie immobile) et un rotor. Deux configurations de rotor sont possibles : interne et externe. Certains moteurs BLDC intègrent des capteurs à effet Hall pour déterminer la position du rotor ; les moteurs à trois fils seulement n'ont pas cette forme de rétroaction. Ils nécessitent un circuit un peu plus complexe et plus de calculs. Pour les moteurs BLDC sans capteur, la note applicative de l'AVR444 [6] est un bon point de départ pour en savoir plus sur le sujet.

La **figure 21** donne la structure d'un moteur BLDC avec rotor externe. Ici, le rotor externe est un aimant permanent. À l'intérieur du rotor se trouvent six enroulements disposés en trois paires. Chaque paire est enroulée de telle sorte qu'un enroulement forme le pôle nord et l'autre le pôle sud lorsque le courant les traverse. Les pôles du rotor tendent à s'aligner avec une paire d'enroulements dans laquelle le courant circule. La **figure 22** montre comment le champ magnétique des enroulements tire le rotor et le fait tourner. Cependant, une seule des trois paires d'enroulements est utilisée à la fois, ce qui peut être amélioré. Quand la paire d'enroulements A attire l'aimant permanent dans la première étape, la paire C peut contribuer à le repousser. Ainsi, deux enroulements participent au mouvement de rotation, augmentant le couple et donc les performances du moteur. La **figure 23** montre la rotation avec deux enroulements alimentés. La chronologie des signaux de commande doit être correcte.

Les enroulements sont entraînés par trois demi-ponts. Vous avez déjà appris à les connaître dans la partie consacrée aux moteurs CC. La

Tableau 2. Séquence pour un moteur pas à pas bipolaire.

	A	B	C	D
1.	1	1	0	0
2.	0	1	1	0
3.	0	0	1	1
4.	1	0	0	1

Figure 19. Moteur pas à pas Joy-IT NEMA17-03 (source : Joy-IT).

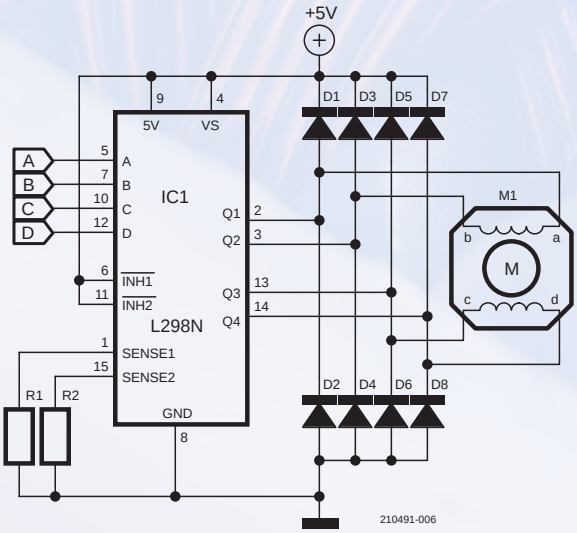


Figure 20. Le L294N en commande de moteur pas à pas.

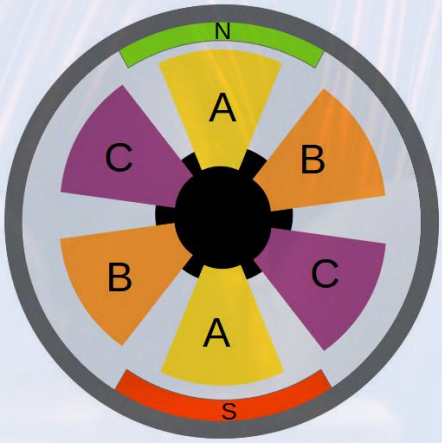


Figure 21. Structure d'un moteur CC sans balais.

figure 24 donne la disposition théorique de ces trois demi-ponts (aussi disponibles sous forme de CI auprès de différents fabricants). Des signaux appropriés doivent être fournis aux entrées A, B et C pour commander le moteur. Le NCP5901B a un diviseur de tension qui polarise chaque broche d'entrée à 2,0 V quand le µC ne la commande pas. Entre 1,3 V et 2,7 V, le NCP5901B se met en mode émulation de diode, c.-à-d. en roue libre, de sorte que l'enroulement n'est pas relié à la masse ni à la tension d'alimentation du moteur. Cela permet par la suite de mesurer la force contre-électromotrice (FCEM) pour déterminer la position du rotor.

Supposons que le rotor tourne et qu'aucun capteur à effet Hall n'est installé dans le moteur, la FCEM de l'enroulement non alimenté peut être utilisée. Cela permet de déterminer la position du rotor, ou plus exactement l'instant où le rotor a tourné suffisamment pour être entre deux paires d'enroulements. Les trois paires d'enroulements A, B et C sont repérées sur la figure 23. Si une tension est appliquée entre A et B, la tension entre les deux enroulements peut être détectée sur l'enroulement C. Au moment où une tension est appliquée entre A et B, un chrono est lancé et la tension sur l'enroulement C est surveillée. Quand celle-ci atteint la moitié de la tension d'alimentation du moteur, le rotor a tourné du demi-angle de rotation entre deux paires d'enroulements. Ce temps est utilisé pour calculer quand les enroulements suivants doivent être alimentés et avec quelle polarité. Par conséquent, l'aimant permanent est maintenu en rotation par le champ magnétique tournant.

La vitesse du moteur peut être contrôlée par des signaux MLI, qui doivent être appliqués en conséquence aux entrées A, B et C du circuit intégré de commande. Le matériel décrit dans la note d'application AVR444 est le kit d'évaluation ATAVRMC100, dont la description et les schémas se trouvent sur la page web de Microchip [7]. Les notes d'application AN1946 de ST [8] et AN12435 de NXP [9] donnent des informations détaillées. La commande de moteurs BLDC nécessite un peu plus de logiciel que celle des moteurs pas à pas ou CC simples. La détection de la position du rotor, le filtrage et le conditionnement interne du signal par logiciel rendent la commande moins intuitive qu'avec les moteurs CC classiques.

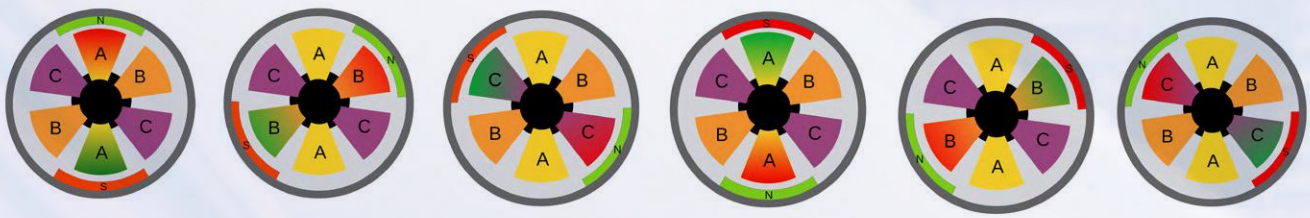


Figure 22. Le rotor est tiré par un champ magnétique.

Figure 23. Le rotor est entraîné par deux des trois paires d'enroulements.

Un pont en H « toutes saisons »

Les ponts en H sont le fondement même des entraînements CC, qu'il s'agisse de moteurs pas à pas, à balais ou sans balais. Il n'est pas nécessaire de les construire avec des composants discrets ; pour des moteurs assez petits et de faibles courants, les CI conviennent s'ils sont disponibles. La commande des moteurs BLDC est plus difficile, et vous devez vous demander si vous voulez acquérir de l'expérience dans la construction de telles commandes ou si vous préférez vous contenter de solutions toutes faites. Cependant, connaître la théorie sous-jacente est certainement profitable.

210491-04

Contributeurs

Idée et texte : **Mathias Claußen**
Rédaction : **Jens Nickel**
Mise en page : **Harmen Heida**
Traduction : **Yves Georges**

Des questions, des commentaires ?

Envoyez un courriel à l'auteur (mathias.claussen@elektor.com) ou contactez Elektor (redaction@elektor.fr).

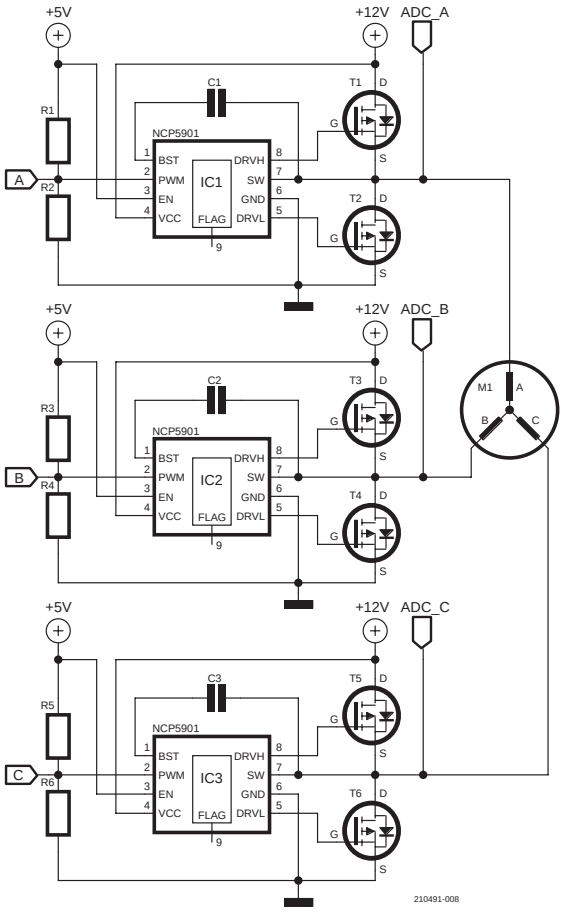


Figure 24. Moteur CC sans balais avec trois demi-ponts.

PRODUITS

- **Kit Geekcreit de 5 moteurs pas à pas 5 V + cartes driver ULN2003 + câbles Dupont**
  www.elektor.fr/19783
- **Commande de moteur CC de Cytron, 3 A, 4-16 V (2 voies)**
  www.elektor.fr/18999
- **Commande de moteur CC de Cytron, 25 A, 7-58 V haute tension**
  www.elektor.fr/19062
- **Shield Arduino de commande de moteur CC de Cytron, 10 A, 7-30 V**
  www.elektor.fr/19063
- **E-book en anglais : « Power Electronics in Motor Drives »**
  www.elektor.fr/18517
- **Livre en anglais : « Book: Motors for Makers »**
  www.elektor.fr/18194

LIENS

[1] Kit de base pour le Raspberry Pi Pico : www.elektor.com/basic-kit-for-raspberry-pi-pico-pico-included
[2] Moteur pas à pas 28BYJ-48 : www.elektor.com/geekcreit-5pcs-5v-stepper-motor-with-uln2003-driver-board-dupont-cable
[3] Cartes de commande de moteur Cytron à pont en H : www.elektor.com/catalogsearch/result/?q=cytron%20driver
[4] Blog de Jangeox : www.jangeox.be/2013/10/change-unipolar-28byj-48-to-bipolar.html
[5] NEMA 17-03 de Joy-IT : https://joy-it.net/en/products/NEMA17-03
[6] AN AVR444 de Microchip : www.microchip.com/en-us/application-notes/an8012
[7] Guide de l'utilisateur du matériel ATAVRMC100 : https://ww1.microchip.com/downloads/en/DeviceDoc/doc7551.pdf
[8] AN1946 de STM : www.st.com/resource/en/application_note/cd00020086-sensorless-bldc-motor-control-and-bemf-sampling-methods-with--st7mc-stmicroelectronics.pdf
[9] AN12435 de NXP Semiconductors : www.nxp.com/docs/en/application-note/AN12435.pdf

l'équipe d'elektor LAB

Notre approche, nos outils préférés et plus encore

Jens Nickel (rédacteur en chef)

Le monde compte de nombreux magazines d'électronique, la plupart se contentent de couvrir l'actualité des innovations et applications. Elektor est différent. Avec son équipe interne d'ingénieurs, Elektor Lab développe, affine et teste en permanence des projets et des produits électroniques.

Depuis six décennies, *Elektor* présente un mélange de théorie et de pratique sain et équilibré. C'est cela son ADN. Tout a commencé il y a plus de 60 ans : Bob van der Horst, passionné d'électronique, trouvait les fiches techniques arides et les articles théoriques trop indigestes. Il manquait un magazine décrivant des circuits que les lecteurs pourraient réaliser et adapter [1]. Dès le début, la plupart des circuits et projets ont été construits et testés avant publication par le magazine *Elektor*. En conséquence, *Elektor Lab* est rapidement devenu célèbre. Le travail du labo d'Elektor perdure encore aujourd'hui. Vous voulez savoir ce qui va se passer au laboratoire d'Elektor en 2022 ? Lisez la suite et rejoignez-nous !

Qu'est-ce qu'Elektor Lab ?

Elektor Lab, c'est à la fois un lieu et une équipe. Chaque membre du labo d'Elektor dispose d'un établi à domicile pour souder, faire de la rétro-ingénierie, programmer et bricoler. Le siège social d'Elektor est aussi doté d'un espace de travail équipé d'un matériel de labo pratique, de composants et d'outils d'ingénierie. Les membres de l'équipe du labo peuvent s'y rendre à tout moment pour tester des montages ou collaborer à des projets.

L'équipe actuelle se compose d'ingénieurs d'âge, de formation et de personnalité divers. Dès le début, le labo a accueilli des analogistes désireux de travailler des mois sur de gros projets, ainsi que des passionnés de programmation voulant apporter de nouvelles idées. En un mot, notre équipe est très diversifiée.

Certains magazines ne parlent que de kits ou de nouveautés. Chez *Elektor*, nous allons beaucoup plus loin. Pour tout nouveau projet, qu'il sorte de notre labo ou de celui de l'un de nos lecteurs enthousiastes, nous nous efforçons toujours d'aller dans les détails, code source compris. Pourquoi cette approche d'*Elektor* ? Notre raisonnement est clair : nous voulons que nos lecteurs puissent construire lesdits projets, et nous les invitons à modifier, améliorer et développer tout ce que nous publions. Que vous soyez ingénieur de profession ou amateur, nous voulons que vous compreniez comment tout fonctionne. Nous sommes fiers de dire qu'*Elektor* est depuis longtemps un des acteurs majeurs du mouvement *open source* !

La plupart des lecteurs d'Elektor et des membres de la communauté savent qu'il n'y a pas de frontière nette entre les ingénieurs de notre labo et la rédaction. Tous les ingénieurs du labo (Clemens Valens, Luc Lemmens, Mathias Claussen et Ton Giesberts) rédigent des articles, et tous les rédacteurs, dont moi-même et le directeur du contenu C. J. Abate, contribuent aux projets et idées du labo.

En ce début d'année 2022, nous souhaitons que vous fassiez connaissance avec l'équipe d'Elektor Lab. Ces pages présentent rapidement les membres de cette équipe et certaines des plateformes et outils que nous utilisons. Nous vous invitons à nous faire part de vos idées, commentaires, projets et articles !

L'équipe d'Elektor Lab

basé en France

Clemens Valens est l'ingénieur qui gère la plateforme Elektor Labs en ligne. Il est titulaire d'un bac scientifique et d'un mastère scientifique en électronique et technologies de l'information. Clemens commença à travailler pour Elektor en 2008 comme rédacteur en chef d'Elektor France. Il travailla également comme rédacteur pour Elektor UK/US et ElektorMagazine.com. Par la suite, Clemens dirigea les labos d'étude d'Elektor aux Pays-Bas, en Allemagne et en Inde. Aujourd'hui, il est le technologue créatif d'Elektor, responsable du site communautaire Elektor Labs où les passionnés d'électronique peuvent publier leurs travaux et interagir avec leurs pairs du monde entier. Il contribue à ses propres projets et à d'autres articles pour le magazine et produit régulièrement des vidéos pour Elektor TV et anime des webinaires. Clemens se passionne pour la synthèse sonore et le traitement du signal.

basé en Allemagne

Mathias Claußen commença par une formation de technicien en électronique des systèmes informatiques et obtint un bac en génie électrique et informatique puis un mastère en systèmes microélectroniques. Il travailla ensuite comme ingénieur projet pour les systèmes embarqués et se spécialisa dans les architectures ARM Cortex-M et les systèmes d'exploitation en temps réel. Mathias rejoint Elektor en 2018. Il se charge principalement des logiciels au sein de l'équipe d'Elektor Lab. Il ne se contente plus de développer des logiciels en arrière-plan. Entre autres choses, Mathias écrit des articles sur le développement matériel et logiciel, et crée des vidéos pour Elektor TV. Si vous voulez en savoir plus sur son espace de travail à domicile, rendez-vous sur Elektor [2]. Vous y verrez des instruments et outils que Mathias utilise quotidiennement.

basé aux Pays-Bas

Luc Lemmens commença à travailler pour Elektor en mars 1990 après ses études à l'Université technique d'Eindhoven. À l'époque, il rédigeait aussi des articles pour *Elex*, un magazine d'électronique destiné principalement aux amateurs débutants. Luc concevait des projets et écrivait pour les deux magazines. Tout l'intéresse, c'est pourquoi aucun domaine de l'électronique ne lui est étranger. Bien sûr, il écrit et publie des logiciels dans de nombreux langages de programmation et, surtout à ses débuts chez Elektor, en assembleur. Aujourd'hui, il se limite généralement à l'EDI Arduino, qui est idéal pour la plupart des projets simples. Au fil des ans, Luc a également écrit et traduit des textes techniques, dont des livres Elektor. Pendant son temps libre, Luc aime jouer au flipper, et surtout réparer et restaurer des machines électroniques modernes et anciennes (électromécaniques, avec relais et unités pas à pas).

basé aux Pays-Bas

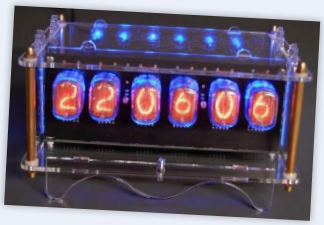
Après ses études, **Ton Giesberts** commença à travailler chez *Elektuur* (rebaptisé *Elektor*) qui recherchait un fan d'audio. Il travailla surtout sur des projets audio durant toutes ces années. Il se passionne pour la conception analogique. Bien sûr, son travail le mène vers d'autres domaines de l'électronique. L'une des devises de Ton est : « Si vous voulez le meilleur, faites-le vous-même ». Prenons le cas d'un circuit imprimé pour de l'audio : pour obtenir une distorsion de l'ordre de 0,001 %, le tracé des pistes et le placement des composants sont cruciaux !

Projets et produits récents d'Elektor Lab

L'équipe d'Elektor Lab a préparé de nombreux montages électroniques pour Elektor. Parfois, une réalisation est entièrement développée par le labo d'Elektor. Les membres de notre équipe adaptent aussi des projets de concepteurs externes. Parfois, une collaboration avec nos partenaires permet d'externaliser des idées. Elle débouchera sur un ou plusieurs articles ou un produit distribué par la boutique Elektor. Voici évoqués ci-dessous quelques projets aboutis récents.

Horloges

Les lecteurs d'*Elektor* ont toujours apprécié les horloges électroniques. Sans doute parce qu'elles sont excellentes pour faire revivre les afficheurs anciens, comme l'horloge Nixie à 6 chiffres [3] et l'horloge « flipper » [4].



Station météo avec ESP32

Achetez un jeu complet et abordable de capteurs météo, ajoutez-y des composants et des logiciels de pointe et construisez votre station météo [5].

CN/A audio pour Raspberry Pi

Les cartes Raspberry Pi ont des sorties audio, mais pour un son de grande qualité, un CN/A externe haut de gamme est nécessaire. Voici la carte [6] conçue par Ton Giesberts, notre magicien de l'audio.

Kit superchargeur LiPo DIY

Vous avez besoin d'un kit d'alimentation LiPo rechargeable pour sortie 5 V et 12 V ? Découvrez le kit développé par Elektor Lab en coopération avec GreatScott! [7]

Nouveau LCR-mètre 50 Hz - 2 MHz

Le LCR-mètre 50 Hz - 2 MHz est un appareil de haute précision pour tester et mesurer les composants passifs. Il peut rivaliser avec les appareils du commerce et même les battre ! [8]

Livre

Maîtriser les microcontrôleurs à l'aide d'Arduino

Le livre de Clemens Valens, *Maîtriser les microcontrôleurs à l'aide d'Arduino*, permet de se familiariser avec le monde d'Arduino et d'apprendre à programmer les µcontrôleurs en général. Parmi les projets proposés, citons un brouilleur de GPS de voiture, une station météo, un télémanipulateur infrarouge et un contrôleur PID.

210545-04 – VF : Yves Georges

Elektor a besoin de vous !

À l'opposé de bien des magazines, les articles et autres contenus intéressants ne circulent pas que dans un seul sens, d'Elektor vers vous. Ils vont également dans l'autre sens. Notre communauté mondiale d'ingénieurs et d'amateurs grandit. Nous l'encourageons à partager ses idées, tutoriels, conseils et astuces, circuits et autres projets.

Vous pouvez enseigner dans un domaine donné de l'électronique ou de la microinformatique. Vous avez conçu un superbe montage électronique. Vous êtes un expert d'un domaine technique particulier… Cela pourrait bien intéresser nos lecteurs. Il y a deux façons de partager conseils, tutoriels et projets :

1) Vous postez quelque chose sur Elektor Labs (voir l'encadré « Plateforme Elektor Labs »). Vous pourrez très vite entrer en contact avec d'autres lecteurs, parfois à l'autre bout du monde, et recevoir un commentaire positif ou critique ! Il est aussi possible que vous ayez des soutiens pour votre projet (voir l'encadré « Elektor Jumpstarter »). Toutes les idées et questions sont les bienvenues sur la plateforme Elektor Labs !

2) Vous pouvez envoyer votre article/projet à l'équipe de rédaction par courriel : editor@elektor.com. Inutile de nous transmettre un article tout prêt, mais un schéma de circuit et une courte description sont indispensables pour nous faire savoir ce que vous avez en chantier. Bien entendu, nous ne divulguerons rien de votre projet sans votre accord !

Les rédacteurs d'Elektor et membres de l'équipe d'Elektor Lab passent en revue les contributions lors de réunions hebdomadaires. Si votre article mérite d'être publié sur papier ou en ligne, nous vous informerons et discuterons avec vous des étapes suivantes.

Plateforme Elektor Labs

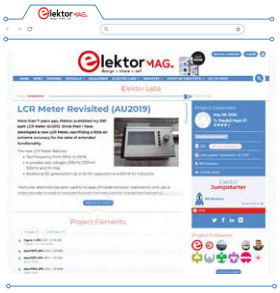
La plateforme en ligne Elektor Labs a été développée pour des professionnels, des amateurs et des étudiants passionnés d'électronique. C'est là que vous pouvez partager vos projets, participer à ceux postés par d'autres, et discuter du développement de projets et d'électronique. Cette plateforme est ouverte à tous, la seule exigence est d'avoir un identifiant Elektor (c'est gratuit !).

Elektor Labs a fêté ses dix ans et a rassemblé plus de 10 000 utilisateurs actifs qui ont posté plus de 2 000 projets et d'innombrables commentaires. La plateforme en ligne est bien plus qu'un espace libre où vous pouvez montrer vos compétences. Non seulement notre équipe de spécialistes peut vous aider à mener à bien votre projet, mais au besoin, elle peut aussi vous aider à en faire un produit qui sera vendu dans la boutique Elektor. Et, comme chaque projet posté est évalué par notre équipe éditoriale, Elektor Labs est également l'endroit idéal pour poster une proposition d'article pour le magazine Elektor.

www.elektormagazine.fr/labs/

Elektor Jumpstarter

Elektor Jumpstarter c'est la version Elektor du financement participatif (*crowdfunding*). Cette plateforme en ligne a pour but d'aider les innovateurs (comme vous) à faire que leurs projets deviennent des produits.



Pour postuler, il suffit de publier un projet sur Elektor Labs, puis de cliquer sur Jumpstarter. Il sera ensuite évalué par Elektor. Dès qu'un projet est accepté, la communauté peut commencer à le soutenir.

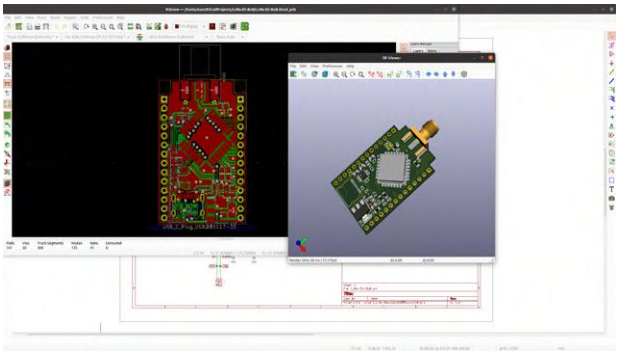
Soutenir un projet n'implique pas d'argent. En revanche, en soutenant un projet, une personne notifie son souhait de pouvoir acheter le produit final si la campagne Jumpstarter atteint son objectif (c.-à-d. une certaine quantité de produits à un prix donné). Si la campagne aboutit, Elektor fabrique le produit et le met en vente dans la boutique.

En bref, chez Elektor Labs, notre devise « concevoir, partager, vendre » prend tout son sens.

www.elektormagazine.com/jumpstarter-info

elektor JUMPSTARTER
from project to product

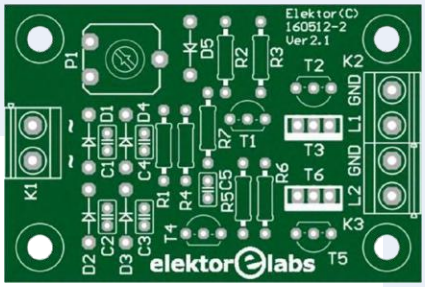
Outil de conception *open source* : KiCad



Elektor aime les solutions *open source*, aussi l'équipe d'Elektor Lab a-t-elle tendance à utiliser des outils de ce type. KiCad en est un bon exemple puisqu'il nous sert (en plus d'Altium et Eagle) à concevoir des schémas et des circuits imprimés. KiCad est gratuit, à code source ouvert et fonctionne sous Windows, MacOS et Linux. Cela permet de partager son travail avec d'autres afin qu'au besoin, ils puissent aussi adapter ou modifier votre création.
Des articles et vidéos sur KiCad sont sur le site d'Elektor : **www.elektormagazine.com/tags/kicad**

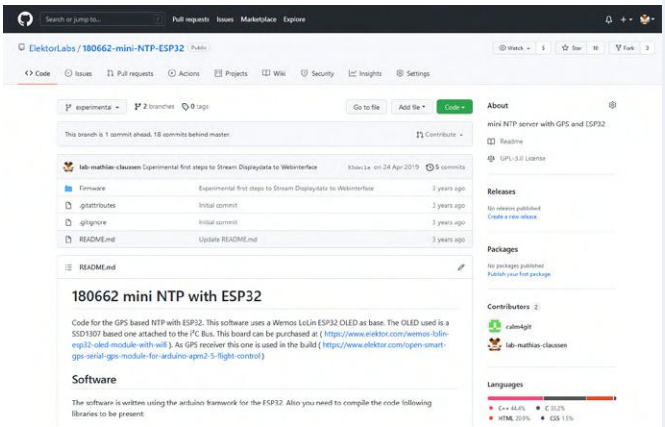
Circuits imprimés Elektor

Les lecteurs d'*Elektor* aiment trouver dans le magazine la reproduction des (petits) circuits imprimés. Pendant fort longtemps, les circuits imprimés étaient disponibles dans notre magasin pour ceux qui voulaient trouver et souder les composants eux-mêmes. En majorité nous n'avons vendu que quelques pièces de ces circuits imprimés nus, et ces chiffres diminuent ces derniers temps. Il faut nous adapter : le monde de l'électronique change. Il n'est plus pertinent de concevoir un circuit imprimé d'ampli audio ou à µcontrôleur (pour n'en citer que deux) de A à Z et de les produire en petites quantités à des prix (trop) élevés. Désormais, amateurs et professionnels ont l'habitude des modules bon marché dont les composants les plus importants (par ex. condensateurs de blocage CMS ou résistances *pull-up*) sont déjà intégrés, c'est donc ce qu'ils attendent. Le temps c'est de l'argent : les développeurs électroniques souhaitent des solutions de prototypage prêtes à l'emploi. Ils veulent ouvrir la boîte, tout brancher et commencer à travailler.
Cependant, cela n'implique pas que nous ne vendons plus de circuits imprimés ou de kits ! Nous misons sur l'intérêt et l'originalité des projets : ceux où nous avons vraiment innové ou inventé et qui ne pourront pas être commandés n'importe où. Pour ces projets, nous collaborons avec des partenaires ayant l'expérience de la production de petites et au besoin de plus grandes quantités. Deux de ces partenaires sont Eurocircuits et SIMAC (Joy-IT).
Si un petit circuit imprimé est nécessaire, mais indisponible via



un partenaire d'Elektor, les fichiers de CAO sont offerts. Chaque lecteur peut adapter le circuit imprimé ou l'utiliser tel quel, et le commander via un service de production. Nous recommandons **www.elektorpcb4makers.com** et **www.elektorpcbservice.com**, que nous proposons avec notre partenaire Eurocircuits.

Elektor @ GitHub



Elektor utilise la plateforme Elektor Labs (www.elektormagazine.fr/labs) pour héberger ses propres projets et ceux d'auteurs externes. Vous y trouverez les fichiers (logiciels et matériels) relatifs aux projets.

Pour les gros projets développés en continu, la gestion des versions successives peut être confuse. C'est là qu'intervient GitHub. Son système de gestion des versions permet facilement de revenir à un état spécifique et de suivre les modifications. GitHub offre aussi d'autres avantages, tels que des versions alternatives expérimentales, le suivi des problèmes pour les suggestions et commentaires, ou des correctifs issus de la communauté.

Par conséquent, nous proposons toujours plus de logiciels et de données CAO à télécharger sur GitHub (https://github.com/ElektorLabs). Le lien web correspondant se trouve à la fin de chaque article de projet.

Pour les débutants de GitHub, Clemens Valens du labo Elektor a créé un tutoriel vidéo sur Elektor TV : **www.youtube.com/watch?v=X5e3xQBeqf8**.

Suivez-nous sur les réseaux sociaux

Membres et rédacteurs d'Elektor Lab sont faciles à joindre sur les réseaux sociaux. N'hésitez pas à nous suivre et à nous contacter sur Twitter, Facebook et YouTube.

Twitter

Mathias Claussen: **https://twitter.com/elektormathias**
Clemens Valens: **https://twitter.com/clemens_elektor**
Elektor: **https://twitter.com/Elektor**
C. J. Abate: **https://twitter.com/elektor_us**

YouTube

ElektorTV: **https://www.youtube.com/user/ElektorIM**

Facebook & Instagram

Elektor Labs (FB): **https://www.facebook.com/ElektorLabs/**
Elektor Labs (IG): **https://www.instagram.com/elektorlabs**

LIENS

[1] « 60 ans d'Elektor », Elektor 05-06/2021 : https://www.elektormagazine.fr/magazine/elektor-176/59580
[2] « L'espace de travail d'un ingénieur d'Elektor pour le développement de logiciels embarqués », site Elektor : https://www.elektormagazine.com/articles/elektor-workspace-embedded-software-development
[3] Horloge Nixie à 6 chiffres : https://www.elektormagazine.fr/180588-03
[4] Horloge rétro à afficheurs de flipper : https://www.elektormagazine.fr/180307-04
[5] Station météo à ESP32 : https://www.elektormagazine.fr/180468-04
[6] CN/A audio pour Raspberry Pi : https://www.elektormagazine.fr/160198
[7] Superchargeur LiPo DIY en kit : https://www.elektormagazine.fr/191188-B-03
[8] Nouveau LCR-mètre 50 Hz - 2 MHz : https://www.elektormagazine.fr/190311-02

Publicité

Une offre encore décuplée

La plus vaste sélection de semi-conducteurs et composants électroniques en stock et prêts à être expédiés™

mouser.fr

MOUSER ELECTRONICS

Raspberry Pi comme télécommande KVM

Test du logiciel Pi-KVM

Mathias Claussen (Elektor)

KVM est l'abréviation de « *Keyboard, Video, Mouse* » (clavier, vidéo, souris). Quiconque y a accès peut commander un ordinateur à distance. Grâce à l'astucieux logiciel Pi-KVM et à un Raspberry Pi 4, vous pouvez commander à peu de frais un PC et d'autres appareils via l'internet, sans avoir à installer de logiciel sur l'ordinateur commandé à distance. Pi-KVM autorise également les disques virtuels, ce qui permet non seulement de commander et maintenir un ordinateur à distance, mais aussi d'effectuer une réinstallation complète.

Pendant la pandémie, j'ai découvert que mes parents avaient encore chez eux des ordinateurs fonctionnant sous Windows 7 (et même sous Windows XP). En raison des failles de sécurité, qui ne peuvent plus être corrigées, ces ordinateurs devaient être mis à jour de toute urgence. Malheureusement, une mise à jour n'était pas suffisante pour ces machines. Elles nécessitaient une réinstallation complète, mais il n'était pas question de se rendre sur place en voiture, en raison du confinement et de plus de huit heures de route.

Quiconque a déjà essayé de guider quelqu'un à travers des dizaines d'étapes d'installation avec un mélange d'appels téléphoniques et de tchat vidéo peut facilement imaginer que ce n'est pas la bonne solution pour l'installation d'un système d'exploitation et la sauvegarde de données. Ce qu'il faut, c'est un accès à distance. Et c'est exactement ce que l'on peut faire avec un RPi comme télécommande KVM. À ce stade du texte, vous pouvez remplacer parents par clients, ordinateurs par commandes de machines et nouvelle installation par analyse des erreurs. Une paire d'yeux avec laquelle on peut observer un système à distance, indépendamment du matériel, permet d'éviter de longs déplacements sur site. Il est également possible de faire fonctionner à distance des appareils et des instruments de mesure qui n'ont jamais été prévus à cet effet par le fabricant. La **figure 1** illustre l'utilisation de Pi-KVM pour le contrôle à distance d'un PC. Une machine peut être maintenue à distance comme le montre la **figure 2**.

L'alternative Pi-KVM

Il est vrai que les logiciels de maintenance à distance existent déjà. Si vous utilisez un RPi ou un PC, vous avez sûrement utilisé des outils comme VNC, AnyDesk ou TeamViewer. En dehors de ceux-ci, il existe nombre d'autres solutions pour la maintenance et la commande à distance. Toutes ces solutions sont peu coûteuses, voire gratuites pour l'utilisateur privé. Pour aider quelqu'un à configurer un nouveau logiciel ou poster des messages sur les réseaux sociaux, c'est une solution simple et rapide. En revanche, ces aides nécessitent toutes un système d'exploitation qui fonctionne et une connexion à l'internet.

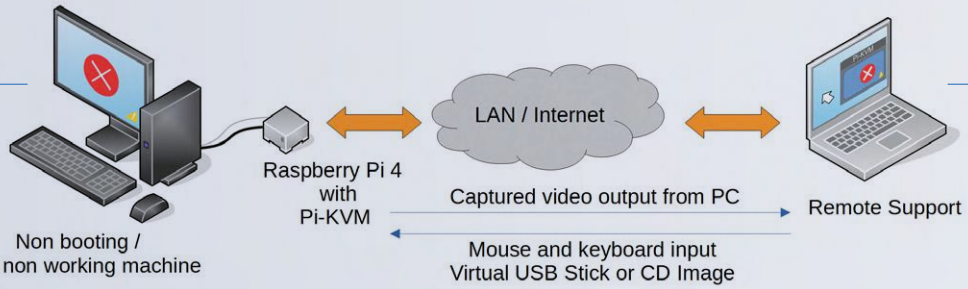


Figure 1. Assistance à distance avec Pi-KVM sur le PC distant.

Mais que faire si l'ordinateur ne démarre plus ou ne parvient plus à se connecter au réseau ? Ou s'il faut réinstaller le système d'exploitation ? C'est à ce moment-là qu'une personne compétente doit se rendre sur place pour faire redémarrer le système d'exploitation et établir une connexion réseau ou internet. Cela s'applique également à la réinstallation d'un système d'exploitation. La **figure 3** montre le message du BIOS d'un PC qui n'a pas trouvé de support de démarrage.

Le Pi-KVM fonctionne indépendamment de l'ordinateur à commander et transmet via le réseau l'image vidéo produite par la carte graphique à un second système qui ne fait qu'exécuter un navigateur web compatible HTML5. Là, non seulement l'image de l'ordinateur commandé à distance peut être affichée, mais les mouvements de la souris et les entrées du clavier sont aussi renvoyés à l'ordinateur asservi. Cela vous permet de commander l'ordinateur distant comme si vous étiez devant son écran, sa souris et son clavier. La **figure 4** montre le message du BIOS du PC maintenant correctement installé, dans le navigateur d'un ordinateur distant. L'accès au BIOS (comme le montre la **figure 5**) peut également être commandé facilement via un navigateur distant, il n'est donc pas absolument nécessaire d'être à proximité du PC.

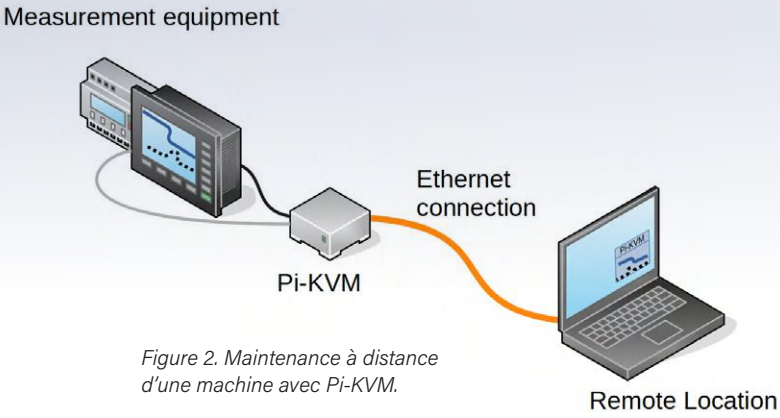


Figure 2. Maintenance à distance d'une machine avec Pi-KVM.

Figure 3. Message d'erreur au démarrage.

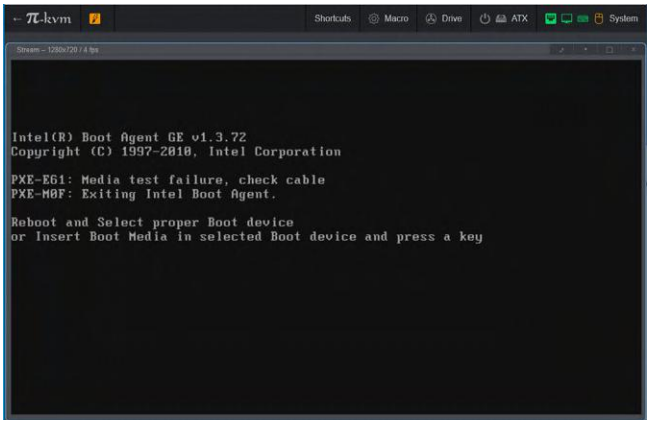


Figure 4. Message d'erreur dans le navigateur avec Pi-KVM.

Disques virtuels

Les choses se compliquent lorsqu'il s'agit d'insérer à distance (typiquement à plus de 2 m) des clés de démarrage USB ou des supports de données rotatifs tels que DVD ou CD. Vous pourriez envoyer une clé de démarrage au PC à commander à distance et espérer que quelqu'un sur place l'insère correctement. Pi-KVM dispose d'une option de disques virtuels (c'est-à-dire qu'il peut émuler une clé USB ou un lecteur de CD-ROM USB). Pour ce faire, il suffit de télécharger et d'installer l'image disque appropriée via un navigateur. La **figure 6** montre le menu du disque virtuel. L'émulation de CD est actuellement limitée aux fichiers ISO, qui ne doivent

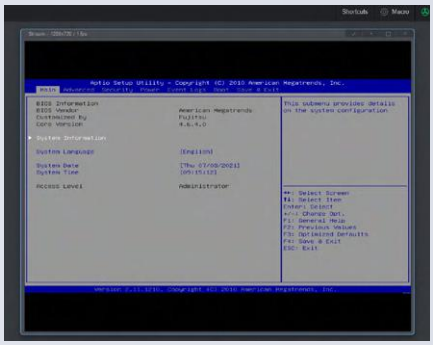


Figure 5. Paramètres du BIOS à distance.

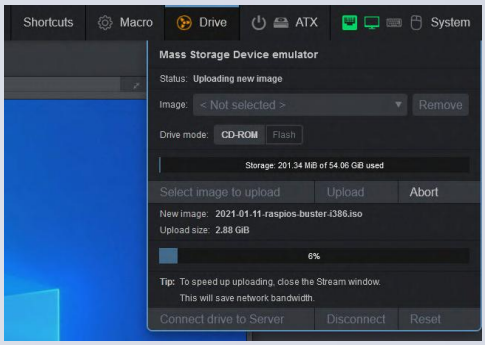


Figure 6. Disques virtuels.

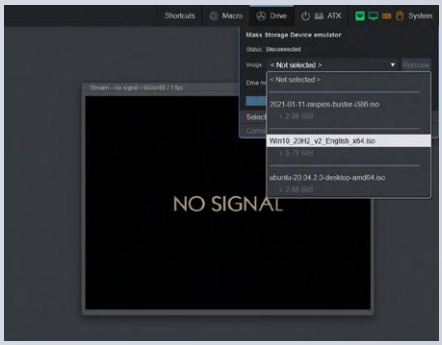


Figure 7. Les ISO de plus de 2 Go comme mémoire de masse.

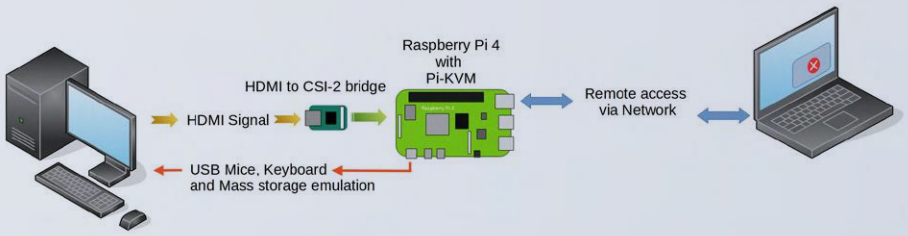


Figure 8. Flux de données avec le pont HDMI-CSI.

Figure 10. Raspberry Pi 4, version 2 Go.

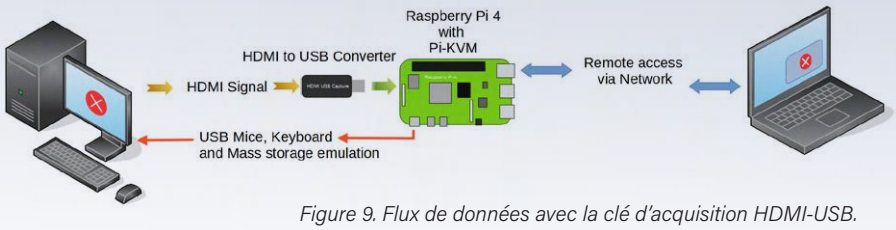


Figure 9. Flux de données avec la clé d'acquisition HDMI-USB.

donc pas dépasser 2 Go. Pour les fichiers ISO de plus de 2 Go (**figure 7**), il faut des ISO hybrides (c'est-à-dire qui peuvent aussi être écrits sur une clé USB). Ceux-ci peuvent alors être utilisés avec le type d'émulation Flash et apparaître comme un stockage de masse USB.

Une solution abordable et ouverte

Est-ce que quelque chose comme Pi-KVM n'existe pas déjà ? Les fonctions offertes par Pi-KVM sont également proposées par des dispositifs d'autres fabricants, mais généralement beaucoup plus chers que la solution Pi-KVM et avec un micrologiciel fermé. Avec ces dispositifs, certaines fonctions de confort font alors également l'objet d'un supplément de prix. Le Pi-KVM est un projet à code source ouvert et peut être assemblé avec du matériel courant. Prévoyez un budget global d'environ 100 € pour la solution faite maison.

Il y a deux façons de construire un Pi-KVM, l'une avec un pont HDMI vers CSI et l'autre avec une carte d'acquisition HDMI vers USB. Les **figures 8** et **9** montrent les flux de données à travers le RPi 4 avec chacune des solutions. Passons maintenant à la présentation des composants requis et à quelques conseils sur leur assemblage.

Matériel nécessaire

Le matériel nécessaire pour le Pi-KVM est assez simple :

- Un Raspberry Pi 4 (2 Go de RAM ou plus) (**figure 10**)
- Une carte Micro-SD (16 Go recommandés)
- Une alimentation USB (5 V/3 A)
- Un boîtier (facultatif)
- Un pont HDMI -> CSI ou une carte d'acquisition HDMI -> USB (clé USB) (**figure 11**)
- Un câble USB en Y (**figures 12** et **13**)

Si les trois premiers composants devraient être disponibles ou faciles à obtenir pour la plupart des lecteurs, les trois derniers sont un peu plus difficiles à trouver. Pour le pont HDMI-CSI (la **figure 14** montre deux variantes), il faut un peu chercher en Europe ou vous rabattre vers l'Extrême-Orient. On peut commander pour quelques euros une clé d'acquisition HDMI-USB auprès de sociétés de vente par correspondance spécialisées. Même si on peut être tenté par le prix de ces clés, elles présentent certaines limites en termes de stabilité ou de résolutions prises en charge. Le pont HDMI-CSI est le choix le plus stable et le plus compatible.

Pour la connexion entre le RPi 4 et l'ordinateur à commander, on utilise le port USB-C du RPi 4 en mode USB OTG. Cela permet au RPi 4 de se comporter comme un clavier, une souris ou un périphérique de stockage de masse vis-à-vis d'un PC. Pour cela, le RPi 4 est connecté à l'ordinateur avec un câble

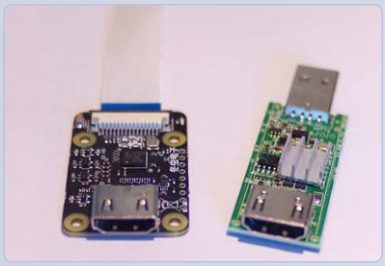


Figure 11. Pont HDMI-CSI et clé d'acquisition HDMI-USB.

Figure 12. Câble USB A-Y.

Figure 13. Câble USB-A vers USB-C.

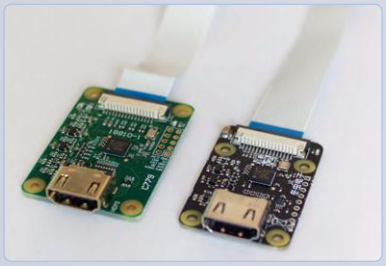


Figure 14. Deux variantes du pont HDMI-CSI.

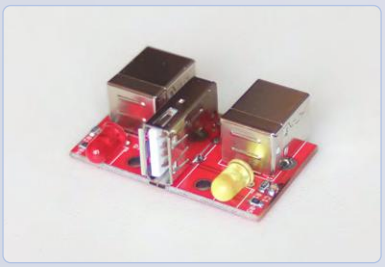


Figure 15. Carte d'adaptation assemblée.

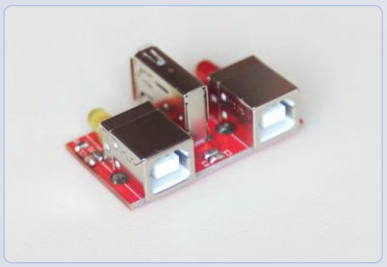


Figure 16. Deux robustes prises USB-B.

Figure 17. Carte micro-SD avec 16 Go de mémoire.

Figure 18. Installation terminée avec le pont HDMI-CSI.

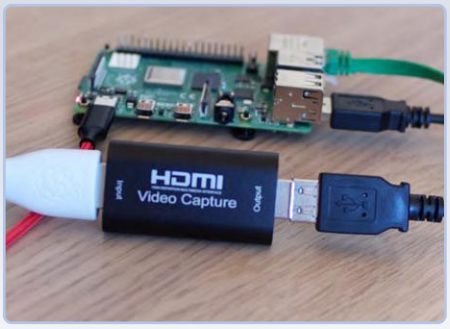


Figure 19. Installation terminée avec la clé d'acquisition HDMI-USB.

USB-A vers USB-C et est également alimenté par ce biais. Comme il faut ici 3 A sous 5 V, on ne peut pas garantir un fonctionnement stable sans un bloc d'alimentation de 5 V supplémentaire.

Le port USB-C étant utilisé en mode OTG, vous pourriez alimenter le RPi 4 en 5 V via l'embase à 40 broches. Cependant, ces 5 V alimentent également l'ordinateur connecté au port USB-C. C'est un phénomène que peu d'ordinateurs apprécient et qui se traduit par des dysfonctionnements. Une solution consiste à utiliser un câble en Y qui sépare les données et l'alimentation électrique.

Pour éviter d'avoir à souder deux câbles USB, le laboratoire d'Elektor a conçu un petit circuit imprimé (**figure 15**). Deux prises USB-B (**figure 16**) permettent une connexion à une alimentation électrique et une autre à un ordinateur sans retour d'alimentation en 5 V. Pour chacune des deux prises, une LED signale la présence de la tension de 5 V du bus. La carte est disponible sous forme d'un projet KiCAD à télécharger dans le dépôt GitHub d'Elektor [1]. Toutefois la carte n'a pas encore été testée de manière approfondie, la réalisation est donc à vos risques et périls. Vous trouverez la liste des composants et le schéma du circuit à la fin de l'article.

Installation du logiciel

Vous pouvez télécharger le logiciel Pi-KVM depuis la page d'accueil du développeur [2]. Une carte SD avec 16 Go de mémoire (**figure 17**) est recommandée pour l'installation. Avec un pont HDMI-CSI, utilisez l'image *Raspberry Pi 4 v2 platform for HDMI-to-CSI bridge* ; avec une clé USB-HDMI, utilisez l'image *Raspberry Pi 4 v2 platform for HDMI-to-USB dongle*.

Le fichier téléchargé doit être décompressé à l'aide d'un programme comme 7-Zip, puis écrit sur une carte SD avec un outil comme Raspberry Pi Imager. L'installation de base est ainsi terminée et vous pouvez commencer à assembler le matériel. Pour examiner le code source, rendez-vous dans le dépôt GitHub [3] de Pi-KVM.

Assemblage du matériel

Si on utilise le pont HDMI-CSI, il est malheureusement un peu plus difficile de trouver un boîtier adapté. Soit quelqu'un a déjà construit un tel boîtier, soit il faut se débrouiller avec un peu de FreeCAD et une imprimante 3D. Pour cet article, le montage est volant.

Dans cette configuration, vous aurez besoin de la carte correspondante et d'un câble de caméra RPi. Il faut insérer dans le RPi 4 une carte SD convenablement préparée avec la dernière image pour Pi-KVM et connecter au port de la caméra le pont HDMI-CSI. Le câble USB en Y doit être connecté au port USB-C du RPi 4. Avec cela, le Pi-KVM est prêt à être utilisé (**figure 18**).

Pour la variante avec la clé d'acquisition

HDMI-USB, il faut l'enficher dans un des ports USB 2.0 du RPi 4. La clé USB ne sera pas reconnue si vous utilisez un des ports USB 3.0 bleus (**figure 19**).

Premier essai

Une fois que tous les composants ont été assemblés et, si possible, installés dans un boîtier, rien ne s'oppose au premier essai. J'ai utilisé un deuxième RPi 4 comme objet de test, celui qui sert depuis un certain temps de caméra sur mon bureau (**figure 20**) (voir l'article du n°491 d'Elektor [4]). Le RPi 4 qui exécute Pi-KVM est connecté à un réseau câblé.

Le premier démarrage de Pi-KVM dure longtemps car la partition est ajustée à la taille de la carte SD. Une fois le processus terminé, Pi-KVM essaie d'obtenir une adresse IP via DHCP. Vous devrez peut-être vérifier dans le routeur ou le serveur DHCP quelle est l'adresse IP du Pi-KVM. Enfin appelez l'interface du Pi-KVM dans un navigateur web.

Après le démarrage, connectez-vous à l'interface avec le nom d'utilisateur *admin* et le mot de passe *admin*. Sous la rubrique KVM, l'écran du deuxième RPi 4 apparaît, la souris et le clavier peuvent maintenant être utilisés à distance.

La **figure 21** montre l'interface web avec le bureau du RPi distant, capturée avec le pont HDMI-CSI. La **figure 22** montre la même scène, cette fois avec une clé HDMI-USB. La qualité de l'image a diminué et des bords verts sont visibles comme artefacts.

Pour des configurations et options supplémentaires, cela vaut la peine de consulter la page d'accueil de Pi-KVM ou le dépôt GitHub. En plus de la souris et du clavier, un stockage de masse est disponible comme déjà mentionné. Le RPi 4 avec Pi-KVM peut émuler un lecteur optique ou une clé USB. Pour ce faire, l'image appropriée peut être copiée sur le RPi 4 via l'interface web (**figure 23**).

Outlook

Avec le logiciel Pi-KVM et quelques composants peu coûteux, vous pouvez transformer un RPi 4 en une télécommande pour ordinateurs et autres dispositifs manipulés avec une souris et un clavier. Si vous ne souhaitez pas assembler les composants vous-même, jetez un œil à la page d'accueil de Pi-KVM. Ils y travaillent sur un HAT complet pour le RPi 4 avec un boîtier correspondant. Pi-KVM et ses développeurs sont actuellement financés par des dons via Patreon ou Paypal. Si vous aimez le projet, l'équipe appréciera sûrement un peu de reconnaissance.

200523-04

Contributeurs

Conception et texte : Mathias Claußen
Rédaction : Jens Nickel
Mise en page : Harmen Heida
Traduction : Denis Lafourcade

Des questions, des commentaires ?

Envoyez un courriel à l'auteur (mathias.claussen@elektor.com) ou contactez Elektor (redaction@elektor.fr).

Figure 20. Caméra Raspberry Pi (Elektor 09-10/2021 [4]).

PRODUITS

- **Raspberry Pi 4 B (4 Go de RAM)**
  www.elektor.fr/18964
- **Carte d'adaptation HDMI -> CSI-2 pour Raspberry Pi (accepte jusqu'à 1080p/25fps)**
  www.elektor.fr/19707

LIENS

[1] Fichiers KiCad sur GitHub : https://github.com/ElektorLabs/200523-Raspberry_Pi_4_with_PiKVM
[2] Page d'accueil de PiKVM : https://pikvm.org/
[3] Dépôt Github de PiKVM : https://github.com/pikvm
[4] Mathias Claußen, « prise de vue et diffusion vidéo en continu avec un Raspberry Pi 4 », Elektor, 09-10/2021 : http://www.elektormagazine.fr/200582-04

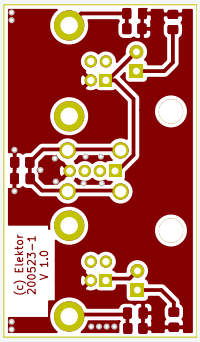
LISTE DES COMPOSANTS

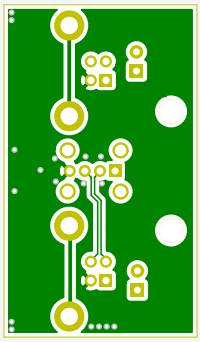
Résistances (0,25 W)
R1, R2, R3 = 100 kΩ
R4, R5 = 1 kΩ

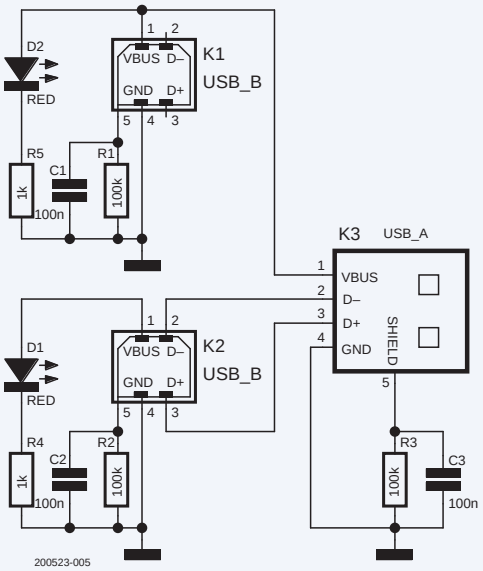
Condensateurs
C1, C2, C3 = 100 nF

Semi-conducteurs
D1, D2 = LED rouge, 5 mm

Autres
K1, K2 = connecteur USB-B
K3 = connecteur USB-A
Circuit imprimé 200523-1







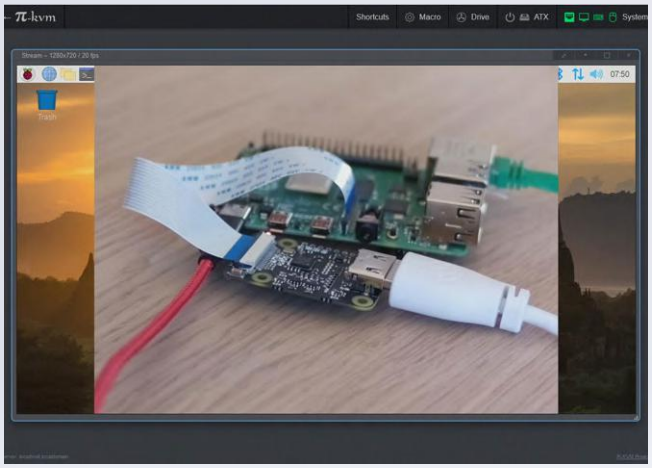


Figure 21. Bureau de la caméra Raspberry Pi capturé avec le pont HDMI-CSI.

Figure 22. Bureau de la caméra Raspberry Pi capturé avec la clé d'acquisition HDMI-USB.

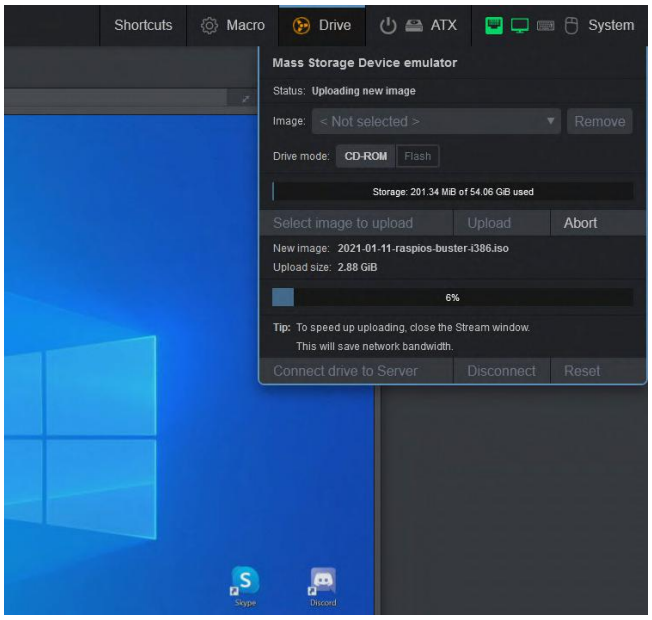


Figure 23. Émulation de mémoire de masse sur l'interface web.

IQaudio **Codec Zero**

carte son au format Raspberry Pi Zero

Mathias Claußen (Elektor Lab)

Vous recherchez une carte son petite et compacte pour Raspberry Pi ? Découvrez l'IQaudio Codec Zero, qui a la taille d'un Raspberry Pi Zero.

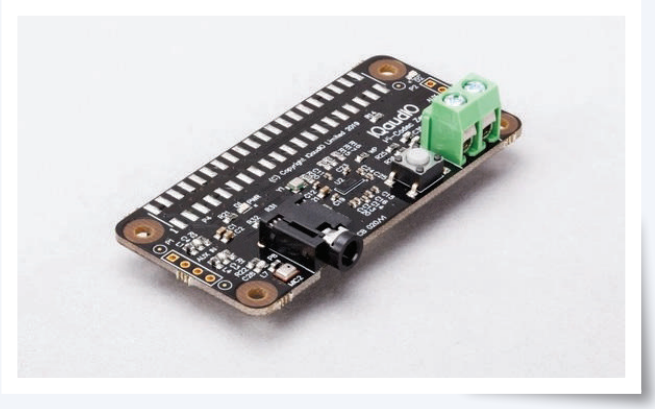


Figure 1. IQaudio Codec Zero. (Source : Fondation Raspberry Pi)

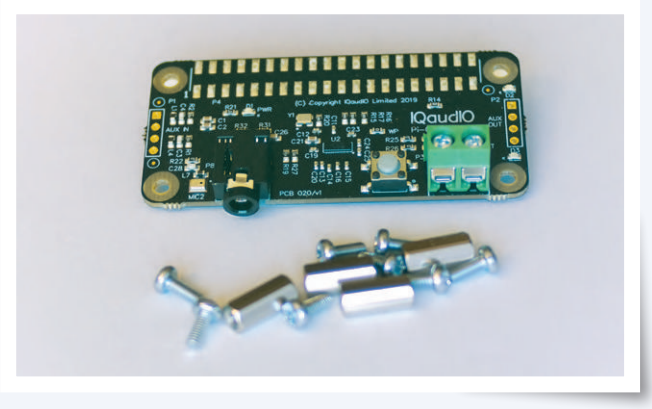


Figure 2. Contenu du kit IQaudio Codec Zero.

Si vous n'avez besoin que d'une sortie sonore numérique sur un connecteur HDMI, vous n'avez pas besoin d'une carte son. Elle devient intéressante si vous souhaitez obtenir une sortie audio analogique de bonne qualité ou si vous souhaitez également traiter des signaux analogiques avec le Raspberry Pi. Les modèles A et B du Raspberry Pi disposent depuis toujours d'une sortie audio analogique sur un connecteur jack de 3,5 mm ; cependant, la qualité n'est pas la meilleure et ne convient pas vraiment à l'écoute de la musique.

L'IQaudio Codec Zero (**fig. 1**) offre un amplificateur intégré de 1,2 W pour un haut-parleur de 8 Ω, un microphone MEMS intégré et une entrée/sortie AUX analogique avec niveau de ligne. Cela permet de réaliser assez facilement des projets tels qu'une enceinte intelligente, un téléphone VoIP ou un jukebox personnalisé. La **figure 2** montre le matériel inclus. Le cœur de l'IQaudio Codec Zero est le DA7212 de Dialog Semiconductor. La **figure 3** montre les blocs et le routage des signaux dans le DA7212. Le circuit intégré est un codec à 24 bits avec une fréquence d'échantillonnage allant jusqu'à 96 kHz. Un égaliseur à 5 bandes, un contrôle automatique de niveau (ALC) et une *noise gate* sont intégrés. La documentation du IQaudio Codec Zero peut être trouvée sur la page de la Fondation Raspberry Pi.

Presque plug & play

L'IQaudio Codec Zero a une EEPROM conforme à la spécification Raspberry Pi HAT, donc un Raspberry Pi devrait automatiquement reconnaître la carte et la configurer de manière appropriée. Sur un Raspberry Pi 3B+, cela fonctionne bien ; le système d'exploitation actuel du Raspberry Pi (32 bits) reconnaît la carte et la configure de manière appropriée. La **figure 4** montre le codec IQaudio dans les paramètres du son. Cependant, il y a encore quelques étapes nécessaires qui ne sont pas directement décrites dans le manuel de l'IQaudio Codec Zero.

Tout d'abord, le volume de sortie doit être réduit au minimum. Après que le codec audio interne du Raspberry Pi ait été désactivé conformément aux instructions, l'IQaudio reste maintenant dans le système comme carte son par défaut. Dans un terminal, un test de haut-parleur peut être lancé avec `speaker-test -t wav -c 1`. Or, le haut-parleur reste muet. Il est nécessaire d'ajuster quelques

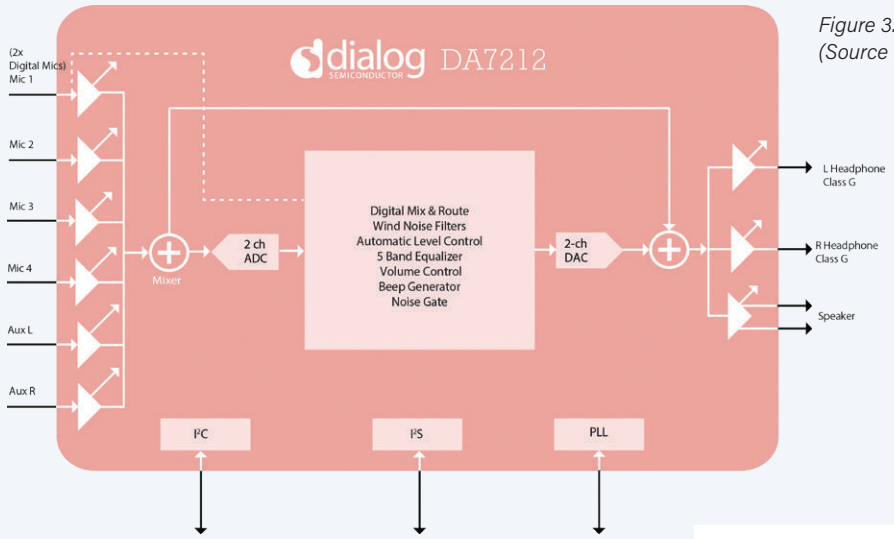


Figure 3. Schéma fonctionnel du DA7212. (Source : Dialog Semiconductor)

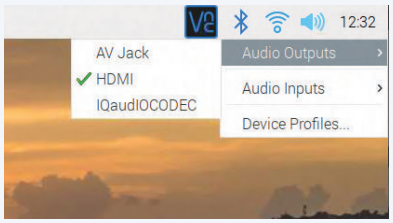


Figure 4. Paramètres du codec IQaudio Zero.

paramètres dans les réglages ALSA de Linux. Le moyen le plus rapide est de recharger les paramètres fournis par IQaudio. Pour ce faire, un dépôt Git est cloné via le terminal avec `git clone https://github.com/iqaudio/Pi-Codec.git`. Dans le terminal, allez dans le dossier créé *Pi-Codec* et exécutez la commande `sudo alsactl restore -f IQaudIO_Codec_Playback_Only.state`. Après cela, `speaker-test -t wav -c 1` devrait produire un son *front left* à partir du haut-parleur. À partir de maintenant tous les sons seront émis par IQaudio Codec Zero. Une installation rapide et facile !

Microphone, E/S AUX

Avec le microphone intégré, le Raspberry Pi peut enregistrer de l'audio. De cette façon, un haut-parleur intelligent peut être réalisé, par exemple, avec *voice2json*. Vous pouvez également utiliser le Raspberry Pi en conjonction avec *Edge Impulse* pour enregistrer des données. Le microphone MEMS de la carte ne gagnera pas de prix dans la catégorie « qualité d'enregistrement », mais il fait l'affaire. Si vous n'êtes pas satisfait du son, vous pouvez connecter un microphone externe. Les fichiers *.state* prédéfinis dans le dossier codec du Raspberry Pi cloné aident à obtenir les paramètres appropriés pour l'enregistrement via le microphone. En utilisant `sudo alsactl restore -f IQaudIO_Codec_OnboardMIC_record_and_SPK_playback.state`, le microphone est sélectionné comme source d'enregistrement. Pour un enregistrement de test, `arecord --device=hw:1,0 --format S16_LE --rate 44100 -c2 test.wav` permet de démarrer depuis le terminal un enregistrement de test avec le microphone ; *CTRL+C* permet de le stopper. Cet enregistrement peut ensuite être lu avec `aplay test.wav`. Si vous voulez connecter une source audio externe, vous pouvez souder des jacks appropriés à l'IQaudio Codec Zero. La **figure 5** montre l'entrée AUX avec des jacks RCA.

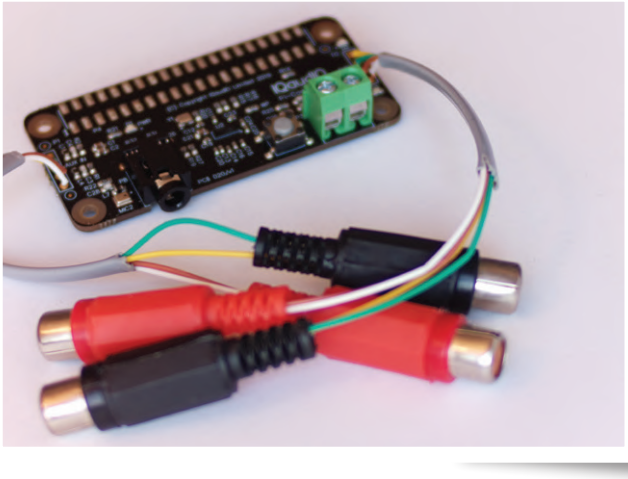


Figure 5. Entrée AUX avec des jacks RCA.

Pour sélectionner AUX In comme source d'enregistrement, vous devez taper ce qui suit dans un terminal : `sudo alsactl restore -f IQaudIO_Codec_AUXIN_record_and_HP_playback.state`
Vous pouvez également voir sur l'image la sortie AUX Out, qui permet à l'IQaudio Codec Zero de servir de source audio à un amplificateur ou à des enceintes actives.

Une nouvelle vie pour une vieille radio

Un élément sur ma liste de choses à faire depuis trop longtemps est la réparation d'une vieille radio Philips (**figures 6 et 7**). Il ne reste dedans que le haut-parleur, et il y a de la place pour un nouveau matériel. À l'intérieur, un Raspberry Pi à faible puissance ferait l'affaire, mais un amplificateur est nécessaire pour le haut-parleur. C'est une bonne chose que le Codec Zero d'IQaudio ait un amplificateur intégré de 1,2 W, de sorte qu'une petite pièce ou chambre puisse être sonorisée à merveille.

Figure 6. Boîtier d'une vieille radio.

Figure 7. Il ne reste qu'un haut-parleur.

Pour la lecture de la musique, on peut utiliser tout ce qui peut se connecter à ALSA. Mais je n'ai pas encore décidé si j'utiliserai le logiciel de lecture audio MPD (Music Player Daemon) ou une distribution spéciale comme Volumio. Je n'ai donc pas encore un concept terminé pour la radio, mais l'un des obstacles, un codec avec un petit amplificateur, a déjà été franchi. Ainsi, au fur et à mesure la radio se remplira pièce par pièce. Il y a encore beaucoup à faire et à planifier pour ce projet.

Une base solide

L'IQaudio Codec Zero offre une entrée et une sortie auxiliaires ainsi qu'un microphone intégré et peut alimenter un petit haut-parleur de 1,2 W. Pour le prix d'environ 20 €, c'est une solution compacte qui peut être utilisée sur un Raspberry Pi Zero ainsi que sur un Raspberry Pi 3B ou 4B. Pour les lecteurs de streaming, les téléphones VoIP ou les enceintes intelligentes, l'IQaudio Codec Zero constitue une base solide. Si vous voulez expérimenter avec l'audio et le streaming sur le Raspberry Pi et Linux, vous devriez considérer l'IQaudio Codec Zero.

210535-04

Des questions, des commentaires ?

Envoyez un courriel à l'auteur (mathias.claussen@elektor.com) ou contactez Elektor (redaction@elektor.fr).

Contributeurs

Texte et illustrations : **Mathias Claußen**
Rédaction : **Jens Nickel, C. J. Abate**
Mise en page : **Giel Dols**
Traduction : **Maxime Valens**

LIENS

[1] IQaudio Codec Zero : https://www.raspberrypi.org/products/iqaudio-codec-zero/
[2] Voice2Json : http://voice2json.org
[3] M. Claussen, « Traitement d'images avec le module Jetson Nano de Nvidia », magazine Elektor 09-10/2021 : http://www.elektormagazine.fr/210318-B-04

PRODUITS

- IQaudIO Codec Zero – carte son pour Raspberry Pi Zero
  www.elektor.fr/19541
- Raspberry Pi Zero WH (avec connecteur soudé)
  www.elektor.fr/18567
- Raspberry Pi 3B+
  www.elektor.fr/18452
- Raspberry Pi 400 – PC à base de Raspberry Pi 4 (version UE) + extenseur GPIO gratuit
  www.elektor.fr/19431

le projet PiKVM et ses enseignements

Entretien avec Maxim Devaev

Mathias Claußen (Elektor)

Installé sur un Raspberry Pi 4, le logiciel PiKVM permet d'accéder à distance à un PC, à un serveur ou à une autre machine par le biais d'une connexion réseau. Maxim Devaev, développeur de PiKVM, nous fait part, entre autres, de ses réflexions sur le projet…

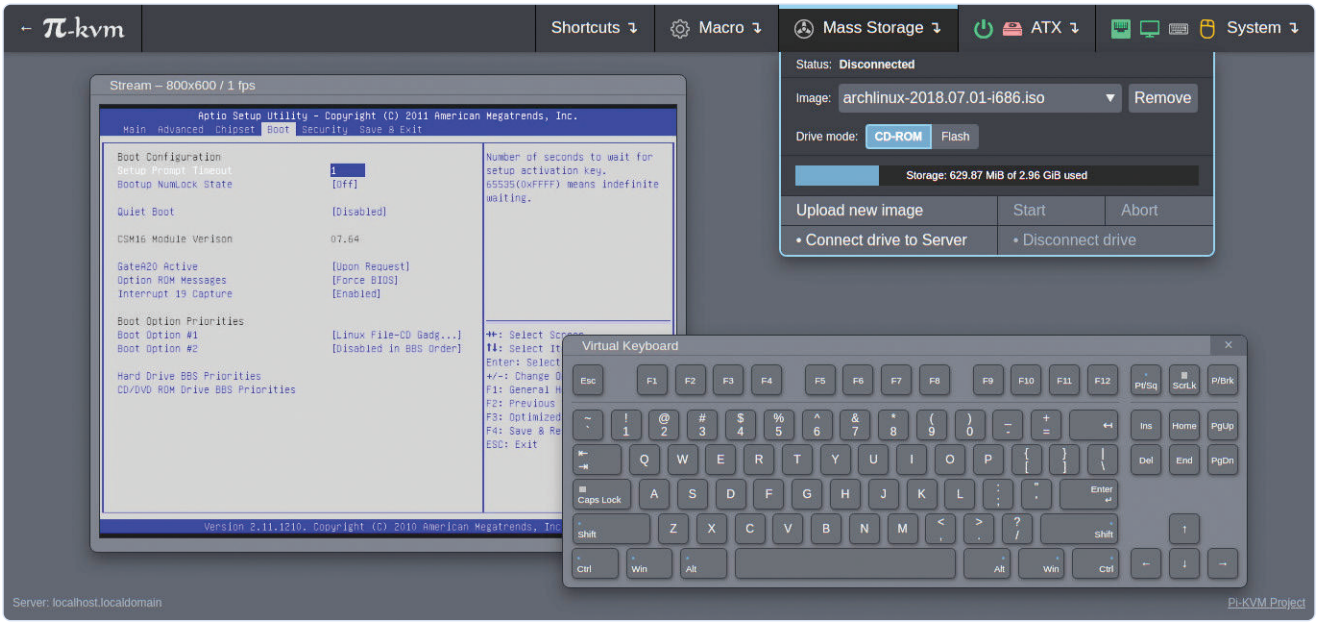


Figure 1. Interface utilisateur de PiKVM.

Le logiciel PiKVM permet de transformer un Raspberry Pi 4 en commande à distance peu coûteuse pour d'autres ordinateurs. Localement, le Raspberry Pi est connecté à l'ordinateur à contrôler. Il extrait l'image du moniteur et émule une souris, un clavier et une clé USB. Les signaux sont ensuite rendus accessibles via l'internet par le logiciel PiKVM exécuté sur le Raspberry Pi – tout ce dont a besoin l'ordinateur de contrôle est un navigateur web. Le projet PiKVM (interface utilisateur illustrée sur la **figure 1**) est *open source* et le coût du matériel requis est inférieur à 100 € pour la version « maison », soit une fraction du prix des produits du commerce. Cela soulève quelques questions que nous avons posées au développeur du logiciel PiKVM, Maxim Deveav.

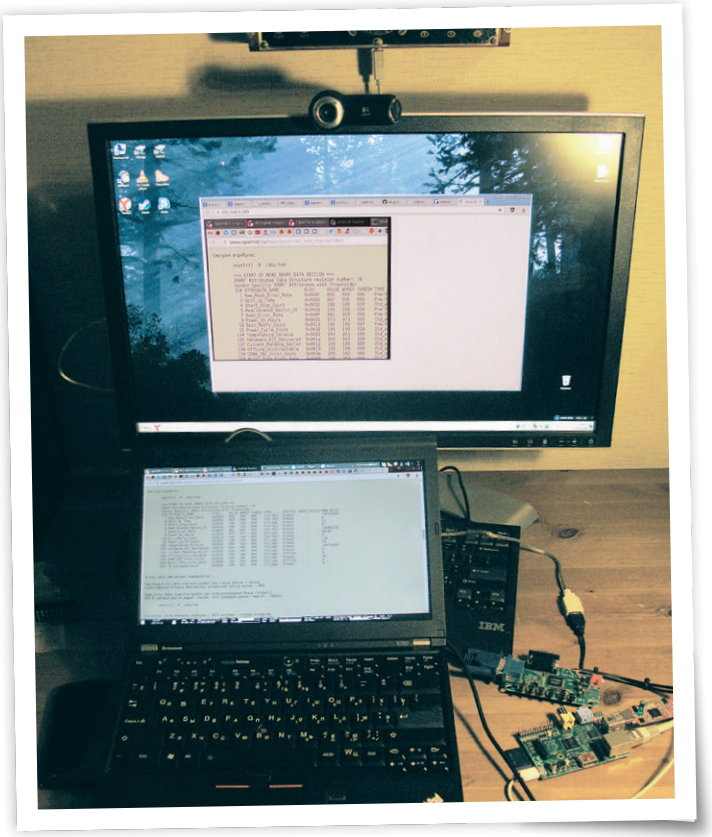


Figure 2. Prototype de PiKVM utilisant un Raspberry Pi 1. (Source : Maxim Deveav)

Mathias : commençons par votre parcours. Comment avez-vous commencé à vous intéresser à l'électronique ? Avez-vous étudié à l'université ?

Maxim : bien sûr, mais il serait inexact de me qualifier de spécialiste en électronique. Je sais comment tenir un fer à souder, et je peux concevoir un schéma de câblage simple et l'assembler sans qu'il prenne feu. Mais je suis avant tout un ingénieur en logiciel. L'électronique est une compétence secondaire pour moi, et je l'ai beaucoup moins développée.

J'ai commencé à m'intéresser à la technique quand j'étais enfant et j'ai lu beaucoup de littérature scientifique et technique depuis. L'acquisition de mon premier PC a bien sûr beaucoup influé sur le contenu de ma bibliothèque, qui s'est étoffée avec tous mes livres d'informatique.

Quelques années plus tard, lorsque le moment est venu pour moi de choisir une université, j'ai opté pour la sécurité informatique. J'ai cependant abandonné peu après, car le programme d'études m'a déçu. L'enseignement supérieur russe est très formel : il est souvent plus important de ne pas échouer à la douzaine de matières humanitaires qui l'accompagne.

J'ai donc suivi une voie d'autodidacte et trouvé un emploi de développeur de logiciels. J'ai travaillé, travaillé, et encore travaillé, je me suis amélioré, et au bout de quelque temps, j'ai atterri chez Yandex (une sorte de Google russe). Je m'y suis installé pendant neuf années entières. Je développais des systèmes distribués, je résolvais de gros problèmes de surveillance et de gestion d'infrastructures, et tout ce qui va avec. Ce n'est peut-être pas l'aspect le plus intéressant, mais c'est ainsi.

Mathias : comment avez-vous eu l'idée de lancer le projet PiKVM ?

Maxim : c'est une histoire amusante, en fait. Une fois, je suis allé rendre visite à mes parents à 1400 km de chez moi. Mon serveur domestique, qui est actif en permanence, est tombé en panne alors que je dégustais le gâteau de ma mère. J'étais ennuyé et j'ai dû demander à mon ami d'aller chez moi pour le redémarrer. C'est ensuite que j'ai décidé d'acheter un KVM (commutateur *keyboard-video-mouse*) sur IP. Je suis allé en ligne, et j'ai vu… 500 $? Sérieusement ? Qu'est-ce qu'il y a dans ce truc pour que cela coûte 500 $? Et puis j'ai lu quelques critiques et j'ai réalisé que, pour 500 $, vous aviez du matériel hors de prix, associé à un horrible logiciel propriétaire bogué, assorti de restrictions de licence. En outre, il n'y avait pas de bons appareils sur le marché : ils étaient tous aussi mauvais les uns que les autres. Je ne sais pas ce qui m'a pris, mais j'ai décidé que, par principe, je ne débourserai pas cette somme. Si vous voulez quelque chose de bien, faites-le vous-même. Et le reste appartient à l'histoire.

Mathias : pourquoi avoir choisi le Raspberry Pi pour ce projet ?

Maxim : le Raspberry Pi est arrivé à point nommé. Il était nouveau sur le marché et convenait parfaitement à un petit système embarqué. En fait, n'importe quel autre ordinateur ARM aurait convenu, mais le prix du nano-ordinateur, sa prédominance et l'excellent support du fabricant ont joué un rôle crucial. Je n'avais pas envie de passer du temps à déboguer des appareils d'origine inconnue, à créer des noyaux personnalisés, etc. Le Raspberry Pi m'a permis de me recentrer sur ma tâche.

Outre le Raspberry Pi, j'avais besoin d'un dispositif de capture vidéo et d'un moyen d'émuler un clavier (je n'avais pas de souris en tête à ce stade). À l'époque, il n'y avait pas de dispositif de capture VGA abordable sur le marché, j'ai donc opté pour un petit bricolage : j'ai utilisé un dongle RCA-USB servant habituellement à numériser de vieilles cassettes VHS, et j'ai introduit la vidéo analogique à partir d'un convertisseur VGA-RCA bon marché (NDLR : voir **fig. 2**). La qualité de la vidéo était horrible, mais de qualité suffisante pour configurer le BIOS et utiliser une console. Quant au clavier, j'ai acheté un convertisseur PS/2 vers USB et j'ai émulé les signaux PS/2 en utilisant le port GPIO. C'était bon marché, primitif, mais ça a marché. Finalement, j'ai abandonné tout ce matériel. J'ai opté pour une passerelle HDMI très diffusée pour la vidéo et j'ai remplacé mon convertisseur PS/2 par un Arduino avec un logiciel personnalisé.

Mathias : comment l'apparition du Raspberry Pi 4 a-t-elle influencé le projet ?

Maxim : outre de meilleures performances et un moyen d'améliorer la qualité vidéo, le RPi 4 disposait de deux contrôleurs USB. Je pouvais utiliser l'un d'eux pour émuler un clavier et une souris dans le noyau, ce qui me permettait de me passer de l'Arduino. J'ai également réussi à construire un lecteur de CD virtuel afin

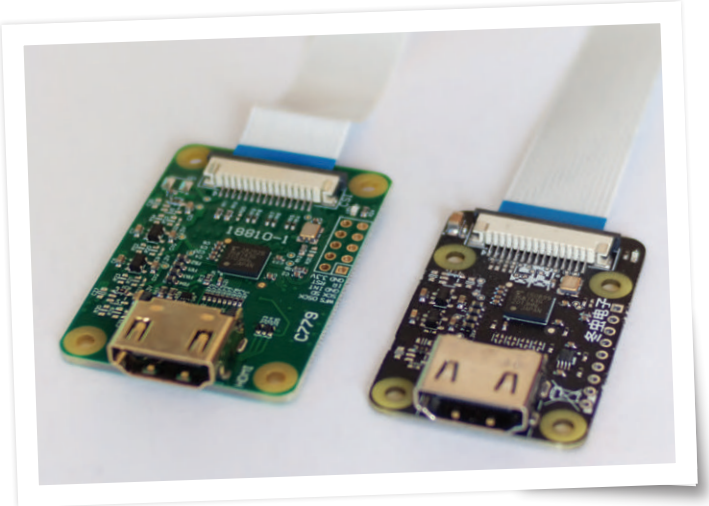


Figure 3. Passerelle HDMI-CSI en deux variantes.

Figure 4. Dongle HDMI-USB.

de pouvoir réinstaller à distance un système d'exploitation sur le serveur. À ce moment, en matière de prix et de fonctions, PiKVM était en avance sur la plupart des dispositifs du commerce.

Mathias : le développement de PiKVM, c'est un hobby ?

Maxim : c'était une sorte de projet personnel. J'ai pris du plaisir à le faire, en me moquant de mon perfectionnisme. Je n'ai jamais pensé que quelqu'un d'autre que moi en aurait besoin jusqu'à ce que je tombe par hasard, sur GitHub (DIY-IPMI), sur un projet similaire à moitié abandonné. Son auteur avait suivi la même voie, mais l'avait laissé en l'état au stade de l'étude de faisabilité. Je me suis enthousiasmé. J'ai commencé à faire de la publicité pour mon projet et à m'engager à fond. J'ai ainsi résolu les problèmes les uns après les autres, ajouté de plus en plus de fonctions et consacré des milliers d'heures au projet. C'était formidable de voir combien de personnes commençaient à utiliser mon logiciel. Auparavant, je n'avais développé que de petits scripts utilitaires ou de très gros logiciels d'entreprise.

Mathias : qu'avez-vous appris de ce projet ? Des expériences à partager avec d'autres développeurs ?

Maxim : oui, en effet. L'interface web est le principal moyen d'utiliser PiKVM, mais il existe aussi un serveur VNC développé en interne. C'est plutôt cool d'utiliser un client VNC pour configurer le BIOS ! J'ai utilisé le format MJPEG pour transférer la vidéo à la fois pour le web et pour VNC. C'est un format simple, mais très peu gourmand en bande passante. Pour le navigateur, j'ai pu utiliser H.264 et WebRTC, mais pour VNC, il n'y avait aucun moyen d'utiliser un encodage différentiel complet.

J'ai pris les choses en main et j'ai écrit au responsable du dépôt TigerVNC et du protocole RFBProto (description de la norme VNC de facto) pour faire une proposition : créons un nouvel encodage, ce serait génial pour les utilisateurs à faible bande passante. Nous avons longuement discuté des détails et nous sommes finalement parvenus à un format qui convenait à tous. Je me suis ensuite adressé à l'IANA pour demander l'enregistrement de cette nouvelle extension VNC. J'avais peur qu'ils refusent, car ils avaient déjà un format enregistré sans description, mais ils ont accepté en expliquant qu'il s'agissait d'un ancien format d'un client propriétaire. Nous sommes en train de développer un *patch* officiel TigerVNC pour la toute première extension *open source* H.264 sur VNC. La version bêta fonctionne sans problème ! Elle est peut-être toute nouvelle, mais certains clients et serveurs VNC sont déjà intéressés pour la mettre en œuvre.

La morale de cette histoire est la suivante : n'ayez pas peur de communiquer avec d'autres projets, même s'ils vous semblent monumentaux. Autre chose aussi : une personne, à elle seule, peut faire avancer le développement de l'ensemble du protocole ; il suffit de rêver grand.

Mathias : l'apparition du dongle USB-HDMI à 10 $ a-t-elle fait avancer le projet PiKVM ?

Maxim : cela dépend. Avant le dongle USB, j'utilisais une passerelle HDMI-CSI (**fig. 3**) qui présentait de nombreux avantages. Le dongle USB (**fig. 4**) a certainement joué un grand rôle dans la popularisation de PiKVM car il était vraiment bon marché. Cependant, j'ai toujours déconseillé de l'utiliser en raison de problèmes matériels : il n'est pas très stable.

De nombreuses personnes ont déclaré l'avoir acheté et l'avoir regretté par la suite, car elles voyaient souvent un écran vide et ne pouvaient rien y faire. Un dongle USB est une boîte noire : vous n'avez aucune idée de son fonctionnement, et si quelque chose ne va pas, vous ne pouvez pas le réparer. La passerelle, en revanche, est beaucoup plus ouverte et presque tout le traitement vidéo est effectué par le noyau et mes logiciels.

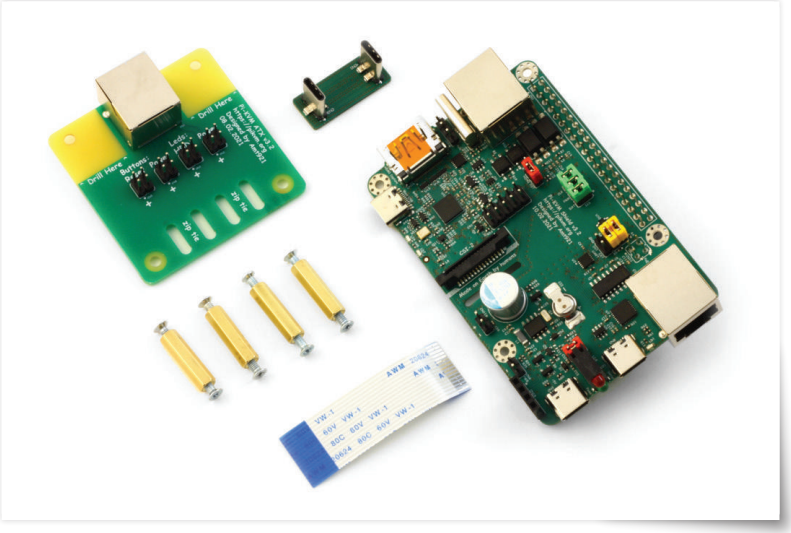


Figure 5. HAT PiKVM v3. (Source : Maxim Deveav)

d'utiliser certaines des fonctions de PiKVM. D'autres créent des rapports de bogues et me donnent un accès à distance à leur matériel pour que je puisse résoudre les problèmes. Mais l'évolution la plus essentielle est ailleurs : lorsque le nombre d'utilisateurs est devenu vraiment élevé, il n'a plus été possible de répondre à chaque personne sur Discord. Les utilisateurs actifs m'ont donc aidé à en intégrer de nouveaux et ils ne me contactent maintenant que lorsqu'ils ne peuvent pas résoudre un problème eux-mêmes. Je suis très reconnaissant envers ces personnes, car je peux maintenant consacrer tout mon temps à l'écriture de logiciels.
Parfois, il y a des personnes qui apportent des correctifs avec de nouvelles fonctions ou des corrections de bogues. Cependant, elles ne sont pas très nombreuses et je suis pratiquement le seul développeur PiKVM pour le moment. J'ai l'intention d'engager des personnes pour m'aider à plein temps avec les logiciels lorsque nous commencerons à vendre des appareils. Ainsi, si vous achetez notre matériel, vous sponsorisez le développement de logiciels *open source*.

Mathias : PiKVM est un projet *open source* que l'on peut utiliser gratuitement (pour l'instant). Comment son développement est-il financé ? Vous semblez maintenant travailler à plein temps sur le projet.

Maxim : je mentirais si je disais que j'ai une vision à long terme, un plan et une bonne compréhension du marché. Tout a commencé l'année dernière, lorsque j'ai quitté mon emploi pour faire une pause et réfléchir à ma carrière. J'ai continué à travailler sur PiKVM comme un hobby. La communauté s'est agrandie et je me suis dit que ce serait formidable si PiKVM devenait mon travail principal. J'ai aussi vu certains projets *open source* dont les auteurs vivaient. Ce serait une situation gagnant-gagnant : je ferais ce qui m'intéresse vraiment, et les utilisateurs obtiendraient un logiciel de qualité en contrepartie d'un petit don. En guise de test, j'ai créé Patreon et j'ai annoncé que j'acceptais les dons, car j'étais désormais sans emploi et prêt à améliorer PiKVM sur demande (au cas où vous auriez besoin rapidement d'une fonction particulière). À ma grande surprise, plusieurs mois plus tard, j'ai commencé à faire quelques bénéfices. C'était bien moins que mon salaire précédent, mais si l'on considère que j'ai arrêté de dépenser de l'argent pour me rendre au bureau, que je dors maintenant suffisamment et que j'ai même perdu un peu de poids, c'était un échange équitable. Les utilisateurs étaient également satisfaits, car ils pouvaient désormais s'écouler une demi-heure entre la découverte d'un bogue critique et sa correction, et je pouvais ajouter de nouvelles fonctions en quelques jours.

Mathias : comment la communauté soutient-elle votre projet, en plus de vous aider à développer à plein temps ?

Maxim : je ne suis pas très doué pour écrire de la documentation, mais beaucoup de personnes m'aident pour cela. Certains écrivent des recettes et des scripts, ou imaginent et décrivent de nouvelles façons

Mathias : si quelqu'un souhaite soutenir votre travail, quelle serait la meilleure façon de le faire ?

Maxim : en dehors des dons, nous apprécions toute aide concernant ce que nous ne pouvons pas encore gérer par nous-mêmes. Certains de ces domaines sont décrits dans nos problèmes GitHub.

Mathias : avez-vous l'intention de commercialiser vos travaux ?

Maxim : actuellement, ma principale source de revenus est constituée de dons. On me suggère souvent que je pourrais vendre une licence commerciale pour le logiciel PiKVM avec des fonctions avancées, mais je ne veux pas le faire pour deux raisons. Premièrement, PiKVM a un accès direct au matériel et je ne ferais pas confiance à un tel dispositif s'il n'était pas *open source*. Deuxièmement, un système à péage conduirait inévitablement quelqu'un à forcer votre logiciel ouvert pour développer des fonctions pro gratuitement. Une partie de votre communauté serait ainsi rattachée à d'autres personnes.
L'autre moyen de monétisation, qui deviendra bientôt le principal, est la vente de mon propre matériel (HAT PiKVM v3) (**fig. 5**), que vous pouvez utiliser à la place de cette version « maison » pour obtenir plus de stabilité et des fonctions matérielles supplémentaires. Mais je n'ai pas l'intention d'abandonner la version « maison » de PiKVM et de forcer les utilisateurs à acheter la v3. Je ne recherche pas la richesse à tout prix, juste assez d'argent pour vivre et nourrir mon chat.

Mathias : pouvez-vous nous en dire un peu plus sur votre matériel, le HAT PiKVM v3 ?

Maxim : pour « bricoler » un PiKVM, vous devez procéder manuellement et prendre éventuellement en main un fer à souder. J'ai décidé de rendre la tâche un peu plus facile pour l'utilisateur, mais

aussi d'améliorer la stabilité matérielle de cette version et d'ajouter certaines fonctions difficiles à mettre au point. Bien que je sois à l'origine du concept original de l'appareil, je ne pouvais pas développer une carte entièrement personnalisée (je ne possède pas ce don spécifique), j'ai donc invité mon ami, compétent dans le domaine du matériel, à travailler avec moi sur ce projet. Notre HAT PiKVM v3 dispose d'un contrôleur d'alimentation de serveur intégré, d'un dispositif de capture vidéo, d'un commutateur USB (pour émuler la sortie d'un câble, c'est parfois utile), et d'un port série (pour se connecter à un serveur ou une console dans un rack), d'une horloge pour une journalisation précise – le tout sur une seule carte. Pour l'essentiel, c'est une solution plus performante et moins chère que toutes celles à 500 $ proposées sur le marché. Vous n'êtes pas obligé de me croire sur parole : les critiques de la v3 sont consultables sur YouTube. Et bientôt, tout le monde pourra l'acheter.

Mathias : vous avez également un boîtier en métal pour le Raspberry Pi 4 et votre module d'extension. Y avait-il une raison de ne pas utiliser de matière plastique ? Nous avons lu sur Discord qu'en Russie, il est plus facile de construire quelque chose en métal qu'en caoutchouc ou en matière plastique.
Maxim : c'est vrai. Le boîtier métallique s'est avéré plus lourd, plus solide et beaucoup moins cher qu'un boîtier en plastique. Ceux qui ont commandé le HAT v3 sur Kickstarter ont pu obtenir ce boîtier gratuitement. Il sera ensuite vendu dans les boutiques en ligne. Et nous avons également un plan gratuit du boîtier pour l'impression 3D, si vous aimez travailler avec vos mains.

Mathias : votre kit sera-t-il disponible pour des clients établis hors de Russie ?
Maxim : le premier lot est maintenant disponible aux États-Unis. Le deuxième lot devrait bientôt être disponible dans le monde entier.

Mathias : quels sont vos projets futurs pour PiKVM ?
Maxim : nous allons continuer à développer à la fois le logiciel et le matériel. Nous prévoyons d'implémenter la capture 60 Hz 1080p (ce qui n'est pas possible actuellement en raison des restrictions du RPi 4) et de créer une solution sécurisée pour l'accès au cloud pour les utilisateurs sans adresse IP externe ouverte.

Mathias : avez-vous d'autres projets en cours de développement ? Sur quoi travaillez-vous ?
Maxim : uStreamer, par exemple. Il a été créé pour PiKVM, mais c'est un composant totalement distinct que beaucoup utilisent comme serveur pour les vidéos de webcam. (Il sert de périphérique de capture dans PiKVM.) Je l'ai écrit à partir de zéro comme une alternative à MJPG-Streamer pour résoudre le problème de performance. Aujourd'hui, uStreamer est le service vidéo MJPEG/H.264 spécialisé le plus rapide, et j'en suis vraiment fier.

Mathias : Maxim Deveav, merci pour cet entretien. Pour ceux qui se dirigent maintenant vers la campagne Kickstarter, nous avons de bonnes nouvelles. La campagne a été plus que réussie. Cela signifie que vous pourrez bientôt commander le HAT PiKVM v3 dans une boutique en ligne. Jetez également un coup d'œil à la boutique Elektor, car notre équipe d'acheteurs travaille actuellement pour pouvoir vous proposer également des HAT PiKVM v3 dans un avenir proche.

210405-04 – VF : Pascal Godart

Des questions, des commentaires ?

Envoyez un courriel à l'auteur (mathias.claussen@elektor.com) ou contactez Elektor (redaction@elektor.fr).

PRODUITS

- **Carte d'adaptation HDMI -> CSI-2 pour Raspberry Pi (supporte jusqu'à 1080p, 25fps)**
  www.elektor.fr/19707

LIENS

[1] M. Claußen, « RaspberryPi comme télécommande KVM », Elektor 01-02/2022 : http://www.elektormagazine.fr/200523-04
[2] Page d'accueil Pi-KVM : https://pikvm.org/
[3] Pi-KVM sur Discord : https://discord.com/invite/bpmXfz5
[4] Problèmes Pi-KVM sur GitHub : https://github.com/pikvm/pikvm/issues
[5] HAT Pi-KVM v3 HAT sur Kickstarter : https://www.kickstarter.com/projects/mMaxim/pikvm-v3-hat
[6] Boutique Elektor : http://www.elektor.fr

véhicule autonome avec lidar 2D

Un ESP32 Pico interprète les données du lidar

Clemens Valens (Elektor)

Avec un lidar qui détecte les obstacles environnants à distance, un robot (ou un véhicule autonome) jouit alors d'une liberté de mouvement bien plus grande. Pour me familiariser avec cette technique, j'ai construit un simple chariot télécommandé et l'ai laissé se déplacer dans mon salon.

Lidar est l'acronyme de ***li**ght **d**etection **a**nd **r**anging* (détection et télémétrie par la lumière). Le lidar est un radar infrarouge (IR), c.-à-d. travaillant dans la partie optique du spectre électromagnétique. La source IR est un laser. Un lidar émet des impulsions IR et mesure le temps qu'elles mettent à revenir en cas de réflexion par un objet distant. La vitesse de la lumière étant une constante connue, la distance de l'objet peut être calculée à partir du temps aller-retour des impulsions (**fig. 1**).

Lidar bidimensionnel

Un lidar peut être unidimensionnel (1D), comme un télémètre laser. Il peut aussi être bidimensionnel (2D), comme un radar de navire ou de tour de contrôle du trafic aérien. Enfin, le lidar peut être tridimensionnel (3D), par ex. sur un avion pour calculer le rendu en relief de la surface de la terre survolée. Ce projet utilise un lidar 2D.

Fondamentalement, un lidar 2D est un lidar 1D rotatif. Au lieu de faire tourner l'ensemble laser/détecteur, il est souvent plus facile de diriger les IR vers un miroir rotatif. Avec des impulsions IR périodiques, le lidar couvre 360° et crée une carte des distances centrée sur lui. La réflectivité des objets est fondamentale. Le corps noir idéal, qui ne réfléchit rien, ne pourrait pas être vu par un lidar.

Connecter le lidar

Pour mon expérimentation, j'ai utilisé le X4 de Ydlidar (**fig. 2**). Il a une portée atteignant 10 m et une résolution angulaire de 0,5° (pour une distance de 50 cm max.). Il embarque un laser infrarouge à 785 nm de longueur d'onde. Il n'utilise pas de miroir rotatif, mais fait tourner l'ensemble laser/détecteur.

Le lidar X4 sort les distances en continu sur un port série à 128 000 bauds. Il est également piloté via cette liaison série. Des commandes simples démarrent et arrêtent le balayage. Il envoie des informations sur demande. Deux fils auxiliaires gèrent séparément les fonctions marche/arrêt et vitesse du moteur. Donc, pour

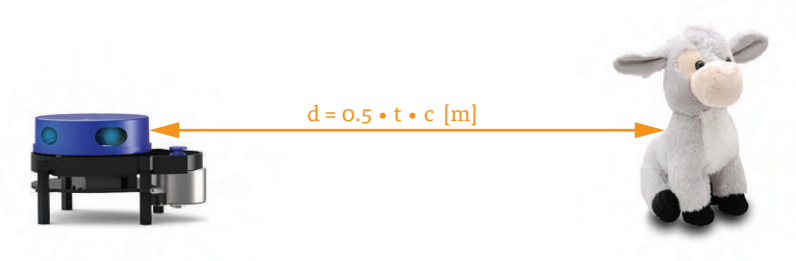


Figure 1. Formule de calcul de la distance. t = temps de parcours de l'impulsion lumineuse en s et c = vitesse de la lumière en m/s. Le facteur 0,5 tient compte de l'aller-retour jusqu'à l'objet c.-à-d. du fait que la distance est parcourue deux fois.

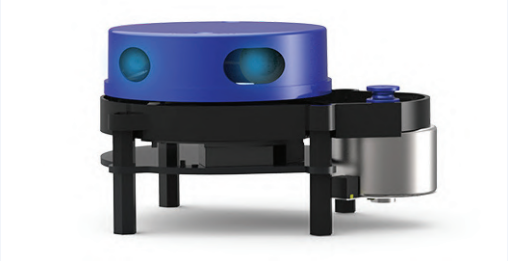


Figure 2. Le YDLIDAR X4 est un lidar 2D peu coûteux dont la portée et la précision sont suffisantes pour qu'un robot se déplace sans heurter d'objets.

que ce lidar fonctionne en mode 1D, il suffit d'arrêter le moteur.
J'ai connecté le port série et les connexions du moteur à un kit ESP32 Pico (**fig. 3**). Pour économiser l'énergie, je n'ai utilisé que la vitesse la plus lente du moteur, soit 400 tr/min. Le lidar consomme alors environ 400 mA.

Analyse des données du lidar

J'ai programmé le kit ESP32 Pico sous EDI Arduino. J'ai écrit une fonction d'analyse des données du lidar, puis vérifié leur interprétation, car le manuel de développement du X4 n'est pas très clair sur le processus et spécifie deux niveaux de détail. Le second niveau offre une meilleure résolution angulaire, mais implique beaucoup de calculs de tangentes inverses, qui accaparent le CPU. J'ai donc d'abord essayé l'interprétation simple.
J'ai placé le lidar sur une table dans un espace rectangulaire clos et je l'ai laissé tourner un moment. Une fois arrêté, je lui ai fait envoyer en valeurs séparées par des virgules (CSV), via un port série, un balayage de 360° moyen que j'ai chargé dans Excel. J'ai pu l'afficher (**fig. 4**) avec la fonction graphique radar. Le résultat fut correct tant en forme (un rectangle) qu'en respect des distances, je n'ai donc pas tenté de l'améliorer en ajoutant des calculs de tangente inverse.

Construire un petit robot

Pour l'étape suivante, j'ai construit un simple chariot télécommandé sur lequel j'ai monté le lidar. Le chariot a deux roues motorisées au centre et une béquille à chaque extrémité.

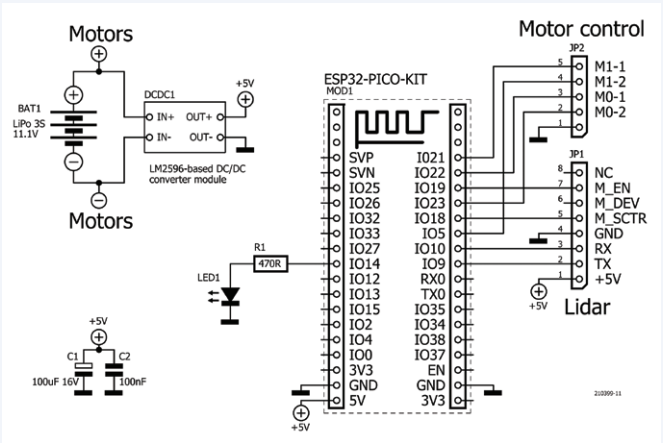


Figure 3. Le kit ESP32 Pico est utilisé comme cerveau du véhicule autonome. La commande des moteurs et le convertisseur CC-CC sont des modules génériques en vente en ligne.

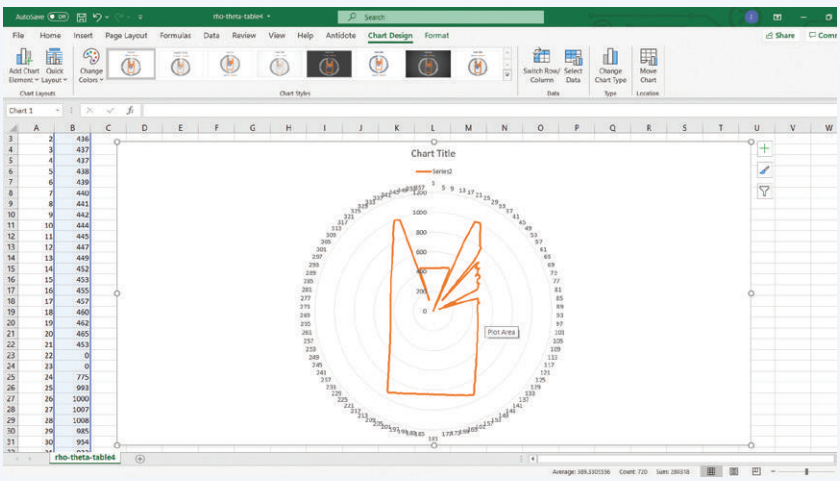


Figure 4. J'ai pu vérifier rapidement mon interprétation des données lidar avec Excel de Microsoft. La courte ligne horizontale à 400 mm (et l'ombre derrière elle) vient de mon ordinateur portable qui enregistre les données du lidar décodées.

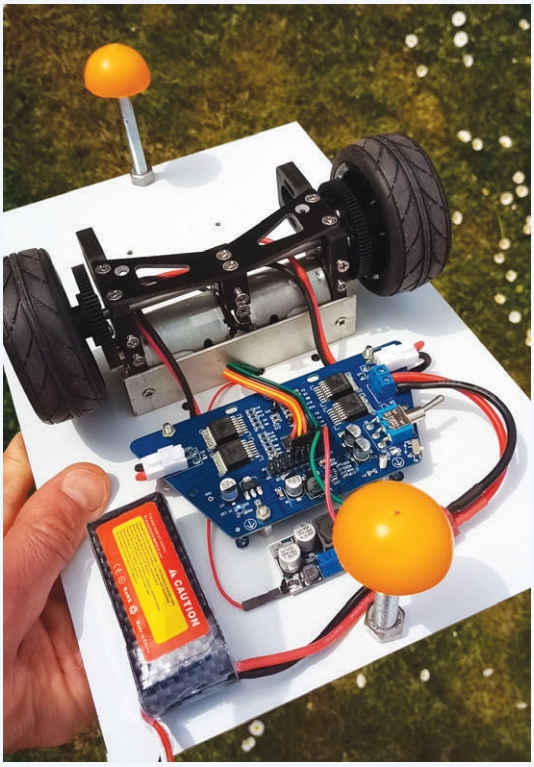
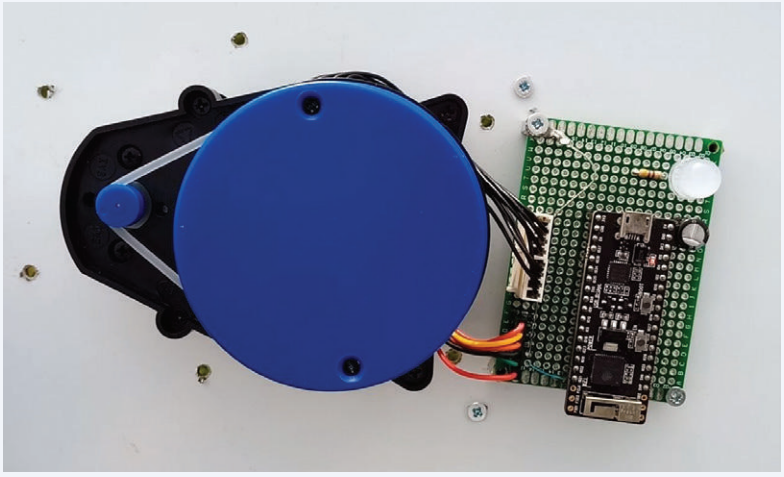


Figure 5. Vue du dessous du chariot. L'ensemble moteur et sa carte de commande viennent de chez Landzo.com mais ne sont plus disponibles.

Figure 6. Face supérieure du véhicule montrant le kit ESP32 Pico et le lidar.



J'ai collé une demi-balle de ping-pong aux béquilles pour réduire les frottements. Tout est monté sur une plaque pour circuit imprimé cuivre double face FR4. La face inférieure accueille les roues avec les moteurs, leur carte de commande et l'alimentation, une batterie Li-Po 3S 11,1 V (**fig. 5**). La face supérieure accueille le module ESP32 et le lidar **(fig. 6**). La plaque est mise à la terre, ce qui protège l'ESP32 des parasites des moteurs. Le centre du lidar et celui de la plaque de montage coïncident avec l'axe vertical du train de roues. Ce chariot simple peut tourner sur lui-même et est assez agile et maniable.

Ajouter une télécommande via Bluetooth

La bibliothèque gratuite et *open source* Dabble [1] fournit le contrôle Bluetooth de l'ESP32 et de l'Arduino, ainsi qu'une application de télécommande pour smartphone avec diverses surfaces de contrôle (**fig. 7)**. L'une de ces surfaces est une manette de jeu, c'était idéal pour mon application. Elle est très facile à utiliser et j'ai pu piloter le chariot avec mon téléphone.

Un algorithme de recherche de chemin

Mon but était de programmer le chariot pour qu'il se déplace tout seul, sans heurter aucun objet, par ex. des meubles. Une approche prisée consiste à laisser le chariot se déplacer et reculer ou s'éloigner dès qu'il s'approche trop près d'un objet, mais cela exige qu'il prenne des décisions. Je voulais quelque chose de plus simple. Beaucoup d'algorithmes simples conduisent à un comportement complexe, par ex. celui d'une nuée d'oiseaux volant ensemble [2]. Je voulais quelque chose de ce genre.

Mon idée fut de faire en sorte que le chariot aille toujours dans la direction de la plus grande distance donnée par le lidar. Pour éviter de tourner en rond, il ne regarde que vers l'avant, dans la plage de −90° à +90°. Il fut assez facile de mettre en œuvre cette règle. À chaque balayage, on met à jour une table avec la distance moyenne à chaque degré. La table comporte donc 360 entrées, une par degré. Dans cette table, on recherche ensuite l'arc de 10° qui donne la plus grande distance moyenne (le choix de 10° relève d'une décision plutôt arbitraire). Le degré médian de cet arc est la direction que le chariot doit prendre. Pour cela, le chariot tourne jusqu'à ce que le degré médian, la bonne direction, tombe à zéro. Il s'agit donc d'un algorithme de commande classique qui tente de minimiser une « erreur » (**fig. 8**).

Premier essai

À ma grande surprise, avec cet algorithme simple, le chariot a réussi du premier coup à se déplacer dans notre salon sans rien heurter (**fig. 9)**. Il a fait le tour du canapé et traversé des passages étroits sans difficulté. Le chariot ne connaît rien de son environnement ni de lui-même, par ex. ses dimensions. De plus, je n'ai tenté aucune optimisation. J'ai juste pris les valeurs que j'ai jugées raisonnables pour chaque paramètre (vitesse de translation, de virage et angle de recherche).

La télécommande Bluetooth est très pratique pour induire le chariot en erreur ou l'aider à se sortir de situations délicates. Elle peut aussi servir à ajuster les paramètres à la volée.

Comme je suis bien davantage motivé par les preuves de concept que par l'optimisation,

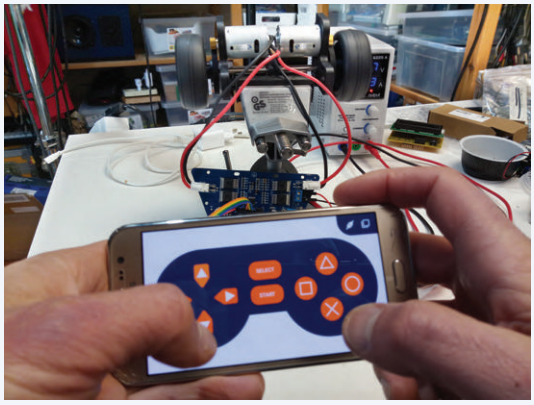


Figure 7. Essai sur établi de la télécommande pour smartphone basée sur Dabble.

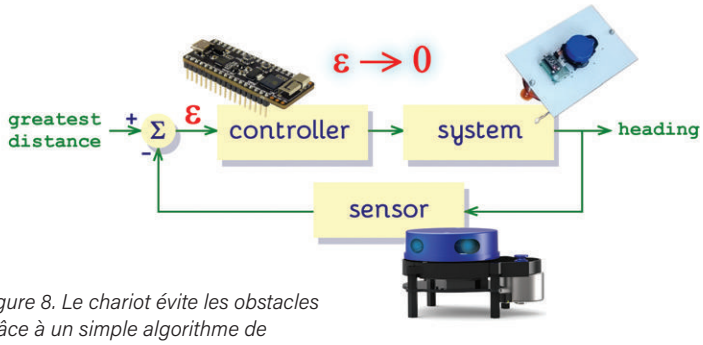


Figure 8. Le chariot évite les obstacles grâce à un simple algorithme de recherche d'un écart minimal.

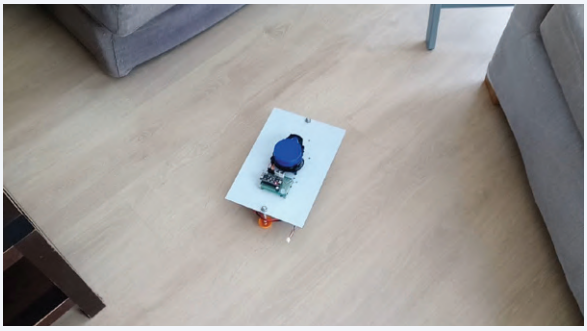


Figure 9. C'est parti ! Regardez la vidéo [4] pour apprécier la fluidité de déplacement.

je me suis arrêté à ce stade. Si la mise au point vous tente, vous trouverez ci-après les liens vers le code. Les possibilités d'amélioration de ce prototype (très loin d'un aspirateur ou d'une tondeuse à gazon autonome) sont nombreuses, mais les résultats obtenus sont très encourageants. Le logiciel de ce projet, un croquis Arduino pour l'ESP32 [3] est téléchargeable. Une vidéo [4] est disponible ainsi qu'un bonus [5].

210399-04

Contributeurs

Idée, conception, texte et photographies : **Clemens Valens**
Rédaction : **Jens Nickel, C. J. Abate**
Mise en page : **Giel Dols**
Traduction : **Yves Georges**

Des questions, des commentaires ?

Envoyez un courriel à l'auteur (clemens.valens@elektor.com) ou contactez Elektor (redaction@elektor.fr).

PRODUITS

- **Kit ESP32 Pico**
  www.elektor.fr/18423
- **YDLIDAR X4**
  www.elektor.fr/18601

LIENS

[1] Dabble : https://thestempedia.com/product/dabble/
[2] Comportement grégaire : https://fr.wikipedia.org/wiki/Comportement_gr%C3%A9gaire
[3] Croquis Arduino ESP32 pour ce projet : https://github.com/ClemensAtElektor/Lidar-controlled-autonomous-vehicle/
[4] Ce projet en vidéo : https://youtu.be/BmNelv_gR9Q
[5] Feux de stationnement basés sur un lidar par Rob Reynolds de SparkFun : https://youtu.be/KRfidalgJx8

le Raspberry Pi Zero 2 W passe au quadricoeur

Mathias Claußen (Elektor)

Le Raspberry Pi Zero, avec son cœur ARM11, a un peu vieilli et ses performances pour de nombreuses applications peuvent être qualifiées de marginales. Heureusement, la Fondation Raspberry Pi a récemment annoncé une mise à jour très attendue, le Raspberry Pi Zero 2 W. Qu'apporte ce nouveau Raspberry Pi ?

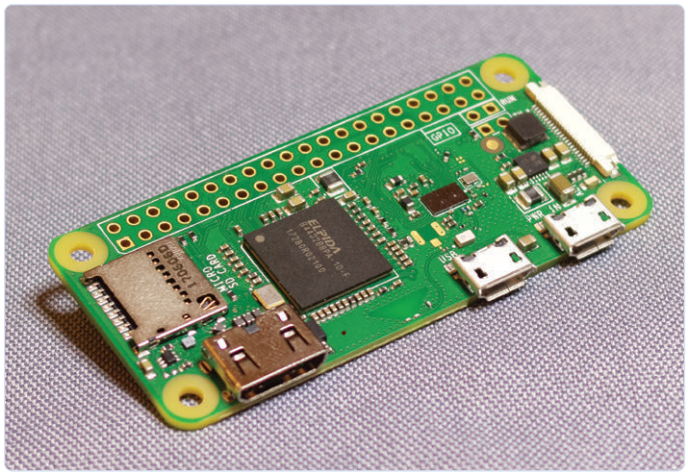


Figure 1. Le Raspberry Pi Zero W.

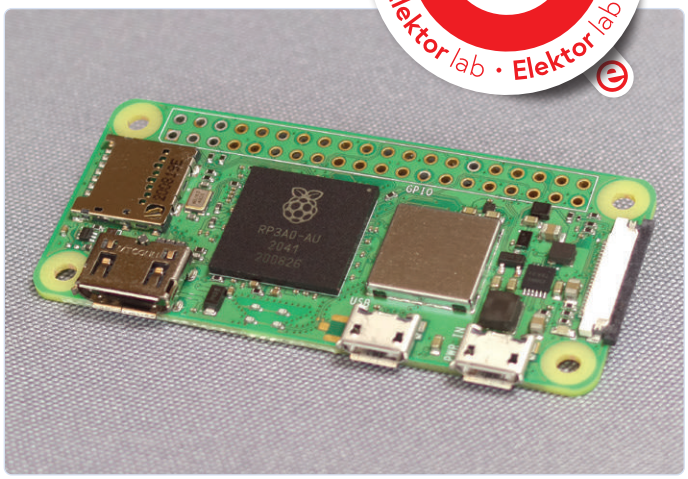


Figure 2. Le Raspberry Pi Zero 2 W.

Le Raspberry Pi Zero W (**fig. 1**) est une petite plate-forme de développement populaire ne coûtant qu'environ 10 €. Avec les périphériques nécessaires, il fonctionne comme un ordinateur autonome, mais il est suffisamment petit et adaptable pour des applications embarquées telles qu'un lecteur multimédia, une caméra ou comme dispositif de commande simple. Lorsqu'il a été présenté en 2017, il s'agissait d'un petit ordinateur compatible avec Linux, alimenté par un SiP monocœur (« **S**ystem **i**n **P**ackage », système dans un boîtier, en français), que nous avons vu pour la première fois intégré dans la version originale du Raspberry Pi 1 (présenté cinq ans plus tôt). La vitesse du développement continu des processeurs est très rapide. Aujourd'hui, le processeur monocœur ARM11, âgé de 10 ans, semble vieux, même s'il fait toujours l'affaire dans de nombreuses applications. Les utilisateurs ont réclamé un remplacement offrant de meilleures performances et une consommation d'énergie plus faible.

À l'automne 2021, les ingénieurs du labo d'Elektor ont eu la chance de mettre la main sur une préversion du Raspberry Pi Zero 2 W (**fig. 2**). C'était une bonne occasion de découvrir sur quoi avaient travaillé Eben Upton et l'équipe Raspberry Pi.

Un remplacement sans préavis

Rien n'a changé concernant le format de la carte et ses connecteurs. La **figure 3** montre une Raspberry Pi Zero W et une Raspberry Pi Zero 2 W, côte à côte. Tous les accessoires et boîtiers du Pi Zero original peuvent être utilisés avec cette nouvelle carte sans aucun problème.

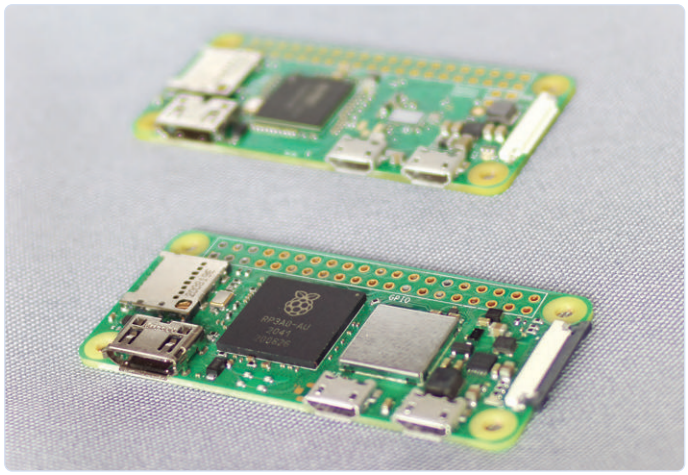


Figure 3. Le Raspberry Pi Zero 2 W et en arrière-plan le Raspberry Pi Zero W.

Tableau 1. Spécifications

- Broadcom BCM2710A1, 4 cœurs Cortex-A53 (ARMv8), 64 bits, SiP cadencé à 1,0 GHz
- 512 Mo LPDDR2 DRAM
- VideoCore IV GPU
- IEEE 802.11.b/g/n Wireless LAN, Bluetooth 4.2/BLE
- Connecteur GPIO à 40 broches
- Port mini-HDMI
- Port micro-USB 2.0 OTG
- Port caméra CSI
- Logement pour carte micro-SD
- Alimentation micro-USB (5 V CC/2,5 A)
- Points de test des broches RàZ et vidéo composite

Comme pour le Raspberry Pi Zero W, nous avons deux ports micro-USB, le connecteur habituel à 40 broches et une sortie mini-HDMI. L'un des deux ports USB est un USB OTG auquel des accessoires USB peuvent être connectés via un adaptateur – ou grâce auquel le Raspberry Pi Zero 2 W lui-même peut devenir un périphérique USB. Comme toujours, une carte micro-SD est nécessaire pour le démarrage. Une vitesse de lecture plus élevée permet de simplifier le fonctionnement du Raspberry Pi Zero 2 W. La **figure 4** montre la face inférieure de la carte qui ne comporte aucun composant.
La **figure 5** montre la face supérieure du Raspberry Pi Zero 2 W, où l'on peut voir que les puces sans fil sont maintenant protégées, et que le SiP nommé RP3A0-AU est la mise à niveau la plus claire. Le contour de ce nouveau SiP est à peine plus grand que celui intégré dans les modèles précédents du Raspberry Pi Zero.

SiP RP3A0-AU

L'étiquette RP3A0 sur le SiP indique « RP » pour Raspberry Pi, et « 3 » pour la génération du SiP (c'est-à-dire qu'il est à base de Raspberry Pi 3). Les spécifications exactes se trouvent dans le **tableau 1**. Pour l'essentiel, le moteur à quatre cœurs du Raspberry Pi 3 A/B+ (d'il y a cinq ans) a été intégré au facteur de forme du Raspberry Pi Zero et tourne à une vitesse d'horloge de 1 GHz. Ceux qui attendaient un Raspberry Pi 4 basé sur le Zero pourraient être déçus, mais sans doute que cela apparaîtra dans une future mise à jour.
Le Raspberry Pi 3A+ utilise un BCM2837B0 cadencé à 1,4 GHz. Dans le RP3A0, on trouve un BCM2837A cadencé à 1 GHz dans le Zero 2 W. Cette fréquence d'horloge plus basse réduit la consommation d'énergie et le dégagement de chaleur. D'autres optimisations de la consommation d'énergie pour les systèmes basés sur Raspberry Pi 3 sont en préparation, comme l'a révélé Eben Upton lors de son interview [1]. L'utilisation d'un SiP de la série Raspberry Pi 3 signifie qu'il y a déjà une base logicielle stable et largement disponible pour l'utilisation sur le Zero 2 W. Les images de la carte SD pour le Raspberry Pi 3 démarreront sur le Zero 2 W sans aucune modification supplémentaire.

Voici quelques-unes des questions importantes que j'ai posées à Eben :

- Comment la consommation d'énergie a-t-elle changé ?
- Puis-je encore utiliser mes anciennes alimentations ?
- Quelles sont les nouvelles fonctions disponibles et qu'en est-il de ses performances ?

Plus petit et moins gourmand en énergie

La faible consommation d'énergie est une caractéristique essentielle de la série Raspberry Pi Zero. Elle est obtenue grâce à une approche

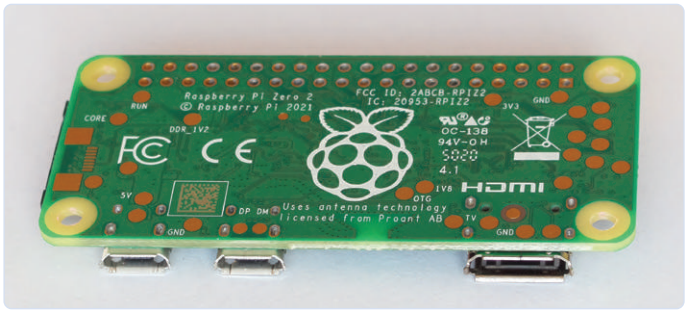


Figure 4. Une vue du dessous du Raspberry Pi Zero 2 W.

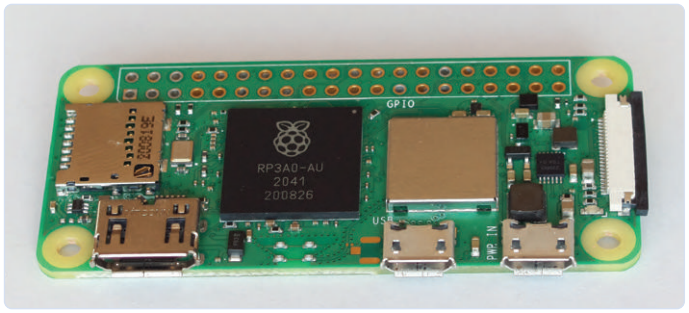


Figure 5. La face « composants » du Raspberry Pi Zero 2 W.

minimaliste de la conception de la carte et au choix de jeux de puces (*chipsets*) bien établis et éprouvés. La consommation d'énergie du Raspberry Pi Zero 2 W est d'environ 2,5 W au maximum avec une sortie HDMI et une liaison Wi-Fi actives. La plupart des ports USB des ordinateurs peuvent fournir jusqu'à 0,5 A sous 5 V, ce qui permet d'alimenter la carte à partir du port USB d'un PC sans avoir besoin d'un adaptateur secteur externe. En mode veille, il consomme environ 0,7 W et les autres modes de fonctionnement réduisent encore la consommation d'énergie. À titre de comparaison, un Raspberry Pi 3B+ consomme environ 2 W en mode veille, ce qui est nettement plus. Sous charge, le Raspberry Pi 3B+ cadencé à 1,4 GHz atteint environ 4,4 W. Le format compact du Raspberry Pi Zero permet de l'utiliser dans une grande variété d'applications, allant de la webradio jusqu'aux consoles de jeux mobiles ou aux systèmes de commande simples. Le nouveau Raspberry Pi Zero 2 W est également idéal comme serveur pour les assistants domestiques ; sa faible consommation d'énergie signifie qu'il a moins d'impact sur l'environnement que les modèles plus grands.

Wi-Fi 2,4 GHz et Bluetooth

Le Raspberry Pi Zero 2 W prend en charge le Wi-Fi 2,4 GHz conformément à la norme 802.11 b/g/n. Bluetooth et BLE 4.2 sont également fournis. Le Wi-Fi dans la bande des 5 GHz est encore réservé à ses prédécesseurs tels que le Raspberry Pi 4B ou le Raspberry Pi 3B+.

Premier démarrage

Lorsqu'il s'agit de démarrer le Raspberry Pi Zero W original, c'est généralement une bonne occasion d'aller prendre un café. Le temps que vous reveniez, tout devrait être stable et prêt à fonctionner. Cependant, avec le Raspberry Pi Zero 2 W, le même processus prend environ 30 s lors du chargement d'une version actuelle de Raspberry Pi OS (32 bits) sans aucune optimisation du chargeur d'amorçage. Le système est alors prêt à être utilisé, tout comme un Raspberry Pi 3. Si vous utilisez un Raspberry Pi Zero original, vous aurez certainement un adaptateur OTG (**fig. 6**) pour connecter une souris et un clavier. Les quatre cœurs du Zero 2 W offrent une augmentation notable du rendement, même si la fréquence d'horloge est la même que celle des versions précédentes du Zero. Dans le Raspberry Pi 3B+, le même processeur tourne 400 MHz plus vite.

Figure 6. Un adapteur USB-OTG.

Les 512 Mo de RAM offrent également un espace suffisant pour certaines applications, mais il ne faut pas s'attendre à ce que le Raspberry Pi Zero 2 W remplace un véritable ordinateur de bureau. Le chargement des pages web devient un peu lourd avec seulement 512 Mo de RAM disponible. Pour cela, un Raspberry Pi 4B ou un Raspberry Pi 400 serait un meilleur choix et offrirait une puissance de calcul considérablement supérieure ; les points forts du Raspberry Pi Zero 2 W résident dans d'autres domaines.

Une mise à niveau modeste, mais bienvenue

Ce dernier Raspberry Pi Zero 2 W est un ajout bienvenu à la famille Zero. Le SiP du Raspberry Pi 3 est certes âgé de cinq ans déjà, mais il s'agit d'une plateforme éprouvée qui représente une amélioration significative de la puissance de traitement du Raspberry Pi Zero. Il bénéficie d'un bon support logiciel ; son SiP Raspberry Pi 3 à quatre cœurs permet à de nombreuses applications logicielles existantes de fonctionner lorsque le besoin en puissance est réduit.
Le facteur d'encombrement du Zero ouvre également de nouvelles possibilités dans des domaines tels que l'automatisation, le rétrogaming ou les lecteurs multimédias. Il s'agit d'une option plus économique pour les projets où un Raspberry Pi 4 est trop cher ou trop gourmand en énergie.

210536-04

Contributeurs

Projet et texte : **Mathias Claußen**
Rédaction : **Jens Nickel**
Mise en page : **Giel Dols**
Traduction : **Asma Adhimi**

Des questions, des commentaires ?

Envoyez un courriel à l'auteur (mathias.claussen@elektor.com) ou contactez Elektor (redaction@elektor.fr).

PRODUITS

- **IQaudIO Codec Zero – carte son pour Raspberry Pi Zero**
  www.elektor.fr/19541
- **Retroflag GPi – Boîtier de type « Game Boy » pour Raspberry Pi Zero (W)**
  www.elektor.fr/19273
- **ZeroDock – Station d'accueil de prototypage pour Raspberry Pi Zero**
  www.elektor.fr/19760

Idée de projet : caméra Wi-Fi avec détection de mouvement

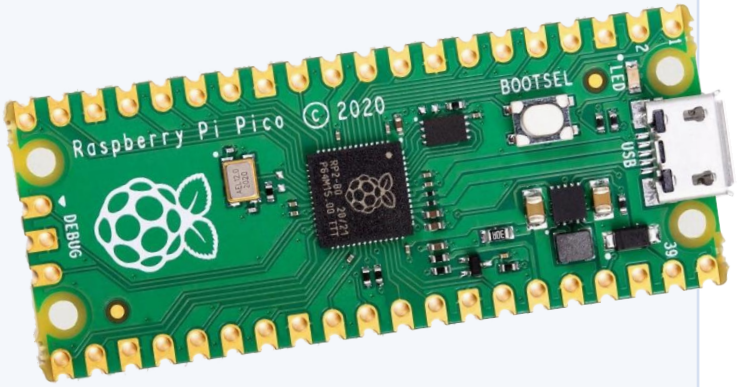
La combinaison d'un Raspberry Pi Zero et du module caméra donne l'une des applications les plus populaires : une caméra compatible Wi-Fi. Le temps de démarrage est cependant assez long et le processeur atteint rapidement ses limites pour la détection de mouvement. Le streaming en direct risque de perdre des images et d'être entaché d'un certain degré de latence des images. Ces effets sont moins flagrants lorsque la même caméra et la même configuration logicielle sont exécutées sur le Raspberry Pi 2 et sur des versions ultérieures plus puissantes.
La puissance de traitement supplémentaire du Raspberry Pi Zero 2 W devrait résoudre les problèmes de perte d'images et de latence et en faire une bonne plateforme compacte pour une utilisation en tant que webcam avec détection de mouvement et diffusion en direct.

Idée de projet : serveur d'imprimante et de scanner USB

Une vieille imprimante peut renaître en tant que périphérique connecté au réseau (dans certains cas sans fil). En utilisant le port USB du Raspberry Pi, il est possible de connecter l'imprimante à un réseau. Selon le fabricant et le modèle, cela peut nécessiter un peu de configuration, mais l'imprimante peut continuer à fonctionner dans son propre réseau avec moins de failles de sécurité. Cela s'applique même aux scanners et aux imprimantes qui ne peuvent plus être utilisés avec les versions actuelles de Windows ou de MacOS [2]. CUPS et SANE sont les représentants classiques de l'impression et de la numérisation sous Linux. Un partageur d'imprimante peut être mis en place en liaison avec Samba. SANEWinDS permet aux clients Windows d'accéder à un scanner via le réseau, de sorte que plusieurs ordinateurs peuvent se le partager. Le Raspberry Pi Zero 2 W peut également être utilisé comme scanner – en utilisant les broches GPIO et quelques scripts, vous serez capable de gérer les traitements complexes de documents.

Idée de projet : lecteur multimédia

Le Raspberry Pi Zero 2 W est idéal pour servir de petit lecteur multimédia. Avec le HAT IQaudio Codec Zero, on obtient un petit appareil de streaming audio. Des distributions comme Volumio sont si faciles à utiliser, tout est supporté de façon optimale, puisque le Zero 2 W est essentiellement un Raspberry Pi 3. Le MPD (*Music Playing Deamon*) peut également être installé sur le RPi Zero 2 W et exploité sur divers appareils avec différents clients. Utiliser le Raspberry Pi Zero 2 W comme clé Wi-Fi avec Kodi est également possible.



Idée de projet : débogueur pour RP2040 et autres microcontrôleurs

La possibilité d'utiliser un Raspberry Pi comme débogueur compatible réseau est quelque peu négligée dans la documentation de la Fondation Raspberry Pi. En travaillant sur les articles pour la sortie du Raspberry Pi Pico, nous avons réalisé nos premières expériences avec un Raspberry Pi Zero en tant que débogueur compatible Wi-Fi. Il faut effectuer quelques ajustements manuels de la configuration, mais le résultat est un débogueur Wi-Fi peu coûteux, non seulement pour le Raspberry Pi Pico, mais aussi pour une large gamme de microcontrôleurs Cortex-M.

LIENS

[1] M. Claussen, « A Decade of Raspberry Pi: An Interview with Eben Upton », Elektor Industry 03/2021 : http://www.elektormagazine.com/210464-01
[2] Serveur d'imprimante et de scanner USB, dépôt Github : http://github.com/sbs20/scanservjs
[3] Raspberry Pi Zero 2 W sur Elektor TV : www.youtube.com/watch?v=3m--7hGiTcE

notes du Forum mondial de l'électronique éthique 2021

Priscilla Haring-Kuipers (Pays-Bas)

Le Forum mondial de l'électronique éthique 2021 – qui s'est déroulé en ligne et en direct au salon productronica le 18 novembre 2021 – a inspiré des concepteurs du monde entier avec un débat public sur l'éthique et les objectifs de développement durable (ODD).

Le 18 novembre 2021, près de 200 personnes se sont réunies virtuellement et physiquement pour participer au premier Forum mondial de l'électronique éthique (WEEF, *World Ethical Electronics Forum*), organisé par Elektor et Elektronik Praxis lors du salon productronica 2021 à Munich. Parmi les orateurs et les participants à cette rencontre poussée par l'industrie, il y avait le sentiment que le moment était venu de voir ce que nous pouvions faire pour rendre l'électronique plus éthique. Au cours de la dernière décennie, de plus en plus de personnes ont réfléchi aux questions d'éthique dans l'électronique. Beaucoup ont discuté des différentes facettes de notre industrie et ont appliqué des pratiques éthiques dans leurs propres entreprises. Le moment semble venu de partager ces réflexions et ces pratiques avec l'ensemble du secteur et d'apprendre les uns des autres.

Regardez WEEF 2021!

Portraits des conférenciers du WEEF 2021

Jetons un coup d'œil rapide à ce que certains des intervenants ont couvert lors de l'événement WEEF 2021. Vous pouvez regarder toutes les conférences du WEEF en ligne sur Elektor TV [2].

- Le professeur Stefan Heinemann a ouvert le WEEF avec son discours d'ouverture intitulé « Focus sur les objectifs de développement durable ». Dans cet excellent exposé, il a encouragé les entreprises électroniques à « faire plus » que ce que la législation actuelle exige. Il a déclaré que « avoir de bonnes mœurs » est une bonne pratique et qu'il ne faut pas écouter les rabat-joies.

- La Dr Paula Palade s'est exprimée sur la manière dont nous pourrions établir une politique autour de la conception et de l'utilisation des voitures à conduite autonome. Elle travaille sur un projet européen visant à rédiger les premières lignes directrices.

- Amir Sherman (Edge Impulse) a souligné l'opportunité d'utiliser l'apprentissage automatique dans du matériel plus petit pour combiner des mesures de la vie réelle avec une modélisation avancée.

- Margot Cooijmans, de la Fondation Philips, a expliqué comment fournir des technologies de santé aux « zones mal desservies ». Cela ne peut se faire en déversant de la technologie gratuite, car cela n'a aucun effet durable sur la santé.

- Dirk Akemann de SEGGER a parlé de l'approche « licence amicale » grâce à laquelle leurs produits sont libres d'être utilisés à des fins privées, éducatives et de prototypage.

- Felix Plitzko, de la société AISLER, s'efforce de « rendre le 'hardware' moins 'hard' » et a promis de rendre possible à l'avenir la production de circuits imprimés neutres en carbone. Il étudie

actuellement cette question en collaboration avec l'institut Frauenhofer [3].

- La famille Brenner (Johannes Brenner et Markus Brenner) s'est jointe à nous pour expliquer comment l'intégration de la pleine conscience au cœur de votre entreprise, plutôt que la recherche du profit, vous permet de prendre soin des personnes qui la composent.

Pour aller plus loin

Bien sûr, nous ne devons pas nous contenter de parler. Si nous voulons être éthiques en électronique, nous devons également agir. Il s'agissait de la première édition du WEEF, et cet événement est appelé à devenir annuel. Le deuxième WEEF est déjà prévu pour novembre 2022. D'ici là, nous travaillerons à l'élaboration d'un annuaire WEEF présentant la plupart des personnes et entreprises éthiques du secteur de l'électronique, ainsi qu'à un manifeste WEEF. Nous aimerions avoir votre avis sur les questions suivantes ainsi que vos suggestions :

- Qui souhaitez-vous voir figurer dans l'annuaire WEEF ?
- Que devrait contenir le manifeste de la WEEF ?
- Quelles sont les questions éthiques importantes dans votre pratique ?

Pour suivre le WEEF et y participer :
www.elektormagazine.com/weef.

210645-04

LIENS

[1] Page web du WEEF : https://www.elektormagazine.com/weef
[2] ElektorTV, « World Ethical Electronics Forum 2021 » : https://youtu.be/ELpxR6SuLb8
[3] AISLER, « AISLER Plans to Ship the First World-Wide Carbon Neutral Circuit Board in 2022 », 18/11/2021 : https://www.elektormagazine.com/news/aisler-plans-to-ship-first-world-wide-carbon-neutral-circuit-board-in-2022

commande de moteur

La commande des moteurs devient simple

Motor Control

Stuart Cording (Elektor)

Le moteur électrique se réinvente sans cesse en raison des nouvelles exigences des applications telles que jouets électroniques, appareils ménagers, outils et véhicules électriques. La commande électronique aussi, pour les mêmes raisons. L'intégration a joué un rôle clé pour obtenir des performances maximales des moteurs. Le pont en H intégré apparu après 1970 révolutionna l'électronique grand public, par ex. celle des magnétoscopes. Aujourd'hui, des microcontrôleurs (*MCU*) optimisés assurent l'efficacité des moteurs sans balais avec des commandes spécialisées.

Les petits moteurs CC à aimants permanents et balais sont omniprésents. Les plus répandus, les moteurs de moins d'un cheval (en anglais *FHP = fractional horsepower*), actionnent les équipements automobiles et les appareils ménagers. Au-delà des essuie-glaces et ventilateurs, les fonctions de confort des voitures font grimper le nombre de moteurs électriques présents : rétroviseurs électriques, lève-vitres, toits ouvrants, portes et sièges réglables. Même s'ils ne constituent pas l'option la plus efficace, les moteurs CC utilisés sont simples à commander sur le plan électrique. Ils restent régulièrement utilisés, sauf dans les applications nécessitant un fonctionnement continu, par ex. pompes et ventilateurs, qui font appel aux moteurs sans balais, plus efficaces.

Commande des moteurs CC à balais

Les moteurs CC à aimants permanents simples ont un stator constitué d'un carter abritant en général deux aimants. Au centre, le rotor, monté sur un arbre, est un induit comportant les enroulements. Ceux-ci sont reliés à un collecteur placé sur ledit arbre. Des balais de carbone, en général solidaires d'un capuchon fixé sur le carter du stator, transmettent l'énergie électrique aux enroulements (**fig. 1**). Le positionnement correct du collecteur sur l'arbre assure un rendement optimal. Idéalement, le plan collecteur-balai, ou plan de commutation, doit faire que le champ magnétique produit par les enroulements du rotor soit perpendiculaire à celui des aimants du stator. Or, en fonctionnement, ce plan se déplace à cause des déformations du champ magnétique du stator. Pour simplifier, on choisit le plan de commutation qui offre le meilleur rendement à un régime donné.

La commutation du moteur étant mécanique, seuls le sens et la vitesse de rotation peuvent être commandés électriquement. S'il n'y a besoin que du réglage de vitesse, il suffit d'un signal modulé en largeur d'impulsion (*MLI*) et d'un organe de puissance de dimension adéquate, tel qu'un MOSFET (**fig. 2**). La plupart des microcontrôleurs (*MCU*) comportent un bloc de temporisations qui, une fois configuré, peut produire un signal MLI sans gestion du processeur. Mieux vaut aussi utiliser un interrupteur *MOSFET* à canal N monté côté masse, car dans les applications à faible consommation, la grille peut souvent être pilotée directement. S'il faut une commutation côté +Vcc, avec un MOSFET à canal N, il faut un circuit auxiliaire pour piloter la grille du MOSFET [1].

Commande du régime et du sens de rotation

Pour asservir le régime, la boucle de commande a besoin du régime instantané

de l'arbre moteur. Une approche consiste à utiliser un capteur, par ex. optique, associé à une roue à fentes solidaire du rotor. Pour un système de commande numérique, la période du signal du détecteur optique peut être convertie en régime moteur (tr/mn). Les MCU disposent en général d'un bloc d'entrées pour capturer des temps sur les fronts montants ou descendants d'un signal. Leur résolution dépend de la vitesse d'horloge du bloc et doit être assez grande pour fournir des données compatibles avec le régime maximal. La sortie de ce bloc alimente un algorithme *proportionnel-intégral-dérivé* (*PID*) [2] qui commande le rapport cyclique de la MLI pour maintenir le moteur à la consigne de vitesse (**fig. 3**).

Toutefois, de tels capteurs montés sur le rotor sont souvent trop coûteux pour l'application, mais on peut aussi utiliser la force contre-électromotrice (*FCEM*) du moteur : proportionnelle au régime, sa précision est plus que suffisante dans la plupart des cas. Pendant les coupures de la commande MLI, on peut mesurer la FCEM directement aux bornes du moteur. Cette méthode est très économique dès lors que le MCU est équipé d'un convertisseur analogique-numérique (*CA/N*). Cette approche ne nécessite en pratique que quelques résistances pour réduire la tension à la plage d'entrée du CA/N. L'implémentation logicielle est plus complexe. La mesure du CA/N et la coupure du MOSFET doivent être synchrones et il faut allouer un court délai de stabilisation de la tension (**fig. 4**). Pour les MCU conçus pour la commande de moteurs, il y aura en général la possibilité d'établir un lien matériel entre la sortie MLI et le déclenchement du CA/N, ainsi qu'un bloc de retard permettant d'automatiser cette approche de mesure.

Une fois les éléments de commande du régime du moteur établis, il ne reste plus qu'à gérer le sens de rotation. Le pont en H (**fig. 5**) utilise quatre N-MOSFET de puissance permettant d'inverser la polarité de l'alimentation des bornes du moteur (soit Q2 et Q3, soit Q1 et Q4). En court-circuitant les bornes du moteur par la paire MOSFET supérieure ou inférieure, on forme un frein électrique qui ralentit le rotor. Là

Figure 1. Les composants d'un moteur CC de moins d'un cheval. Le collecteur est bien visible sur le rotor, près des bobines de l'induit. (Source : ShutterStock/Pixel Enforcer)

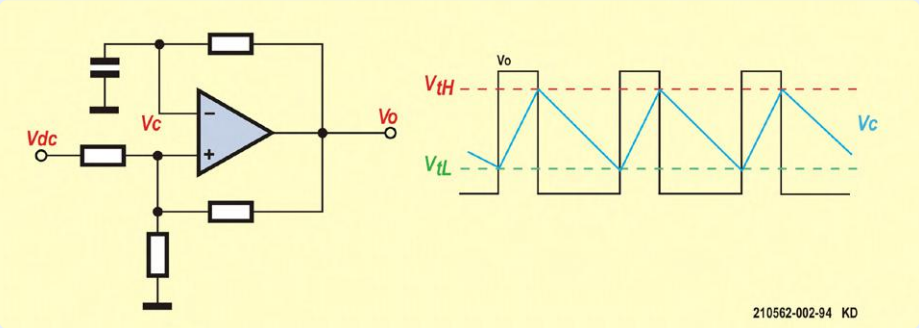


Figure 2. Générateur MLI basé sur un ampli opérationnel.

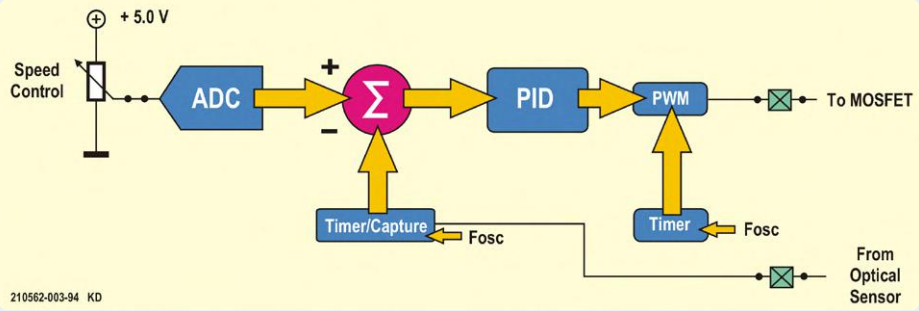


Figure 3. La capture de données temporelles est possible en sortie d'un capteur solidaire du rotor. Combinées avec la consigne de vitesse, elles sont transmises au PID pour réguler le régime moteur. (Source : Microchip Technology)

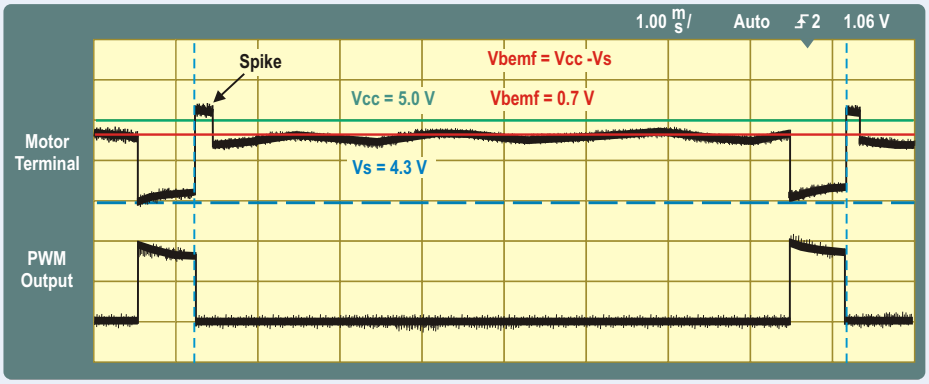


Figure 4. Après la coupure du MOSFET, un petit pic apparaît aux bornes du moteur. Les mesures du CA/N doivent attendre que la FCEM ait atteint un état stable. (Source : Precision Microdrives)

encore, une MLI sur l'un des MOSFET (ou les deux) pourra asservir la vitesse. Il faut toutefois noter qu'il existe un risque de court-circuit de l'alimentation en engageant les deux MOSFET simultanément.

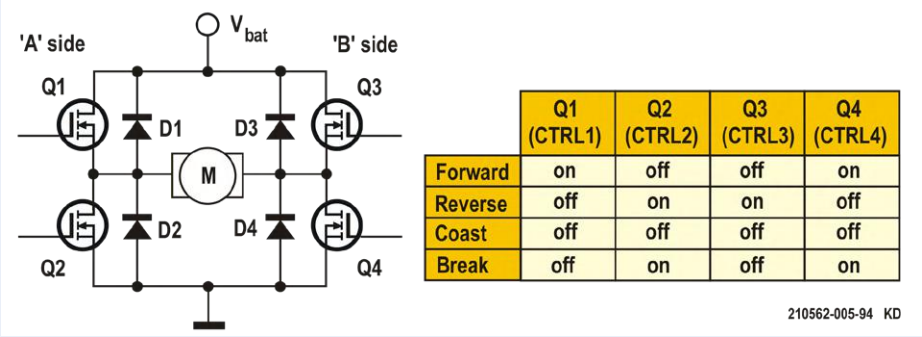
La plupart des fournisseurs de circuits intégrés (CI) proposent des ponts en H à haute intégration, simples d'utilisation avec une multitude de fonctions de protection et de diagnostic. Par ex. l'*IFX9201SG* [3] d'Infineon qui, avec un courant de 6 A, vise la commande de moteurs en milieu industriel. Ce CI s'interface à des E/S à usage général et à un signal MLI, afin de commander la vitesse et le sens de rotation. En parallèle, une interface SPI offre l'accès aux registres de diagnostic. Ceux-ci renvoient les données de courts-circuits par rapport à Vcc et à la masse, les conditions de charge ouverte et les sous-alimentations. Au besoin, la vitesse et le sens de rotation peuvent se piloter par SPI plutôt que par les broches d'entrée de sens et de MLI. Le risque de court-circuit involontaire du pont en H est ainsi éliminé, et cela inclut des protections supplémentaires contre la surchauffe et la surintensité. Pour prototyper rapidement une étude, vous pouvez acquérir le kit *H-Bridge 2GO* [4] qui s'intègre à *DAVE*, l'environnement de développement logiciel d'*Infineon*.

Optimisation du régulateur PID

La sélection d'un MCU répondant au niveau de prix souhaité pour l'application aboutit souvent à un MCU à 8 bits. L'inconvénient de ces MCU réside dans leurs piètres performances arithmétiques. En conséquence, un régulateur PID aura du mal à suivre la cadence des changements de consignes de vitesse et des variations de charge. Pour pallier cet écueil, certains MCU PIC intègrent un bloc accélérateur PID (**fig. 6**) à accumulateur à 35 bits, comme le *PIC16F1619* de *Microchip*. Grâce à l'équation standard du domaine *s* (fréquence complexe) du PID, on écrit trois constantes (K1, K2 et K3) dans les registres du module puis le point de consigne souhaité et on lance le module de calcul PID. Le module PID réduit à 9 le nombre de cycles d'horloge requis, contre 1000 environ pour un PID logiciel sur ledit MCU [5].

Avantages des moteurs BLDC et PMSM

Si la simplicité des moteurs CC FHP est attrayante, d'autres caractéristiques, comme l'efficacité, les nuisances sonores et la densité de puissance accrue, orientent de plus en plus les applications vers les moteurs CC sans balais (*BrushLess DC* = *BLDC*) et synchrones à aimants permanents (*Permanent Magnet Synchronous Motors* = *PMSM*). Avec un pilotage optimal, les BLDC et les PMSM sont bien plus efficaces et dissiper la chaleur est plus facile, car les enroulements sont dans le stator.



	Q1 (CTRL1)	Q2 (CTRL2)	Q3 (CTRL3)	Q4 (CTRL4)
Forward	on	off	off	on
Reverse	off	on	on	off
Coast	off	off	off	off
Break	off	on	off	on

Figure 5. Circuit de pont en H avec tableau décrivant comment la commutation commande le sens de rotation. En court-circuitant les bornes du moteur, on peut mettre en œuvre un frein électrique. (Source : Microchip Technology)

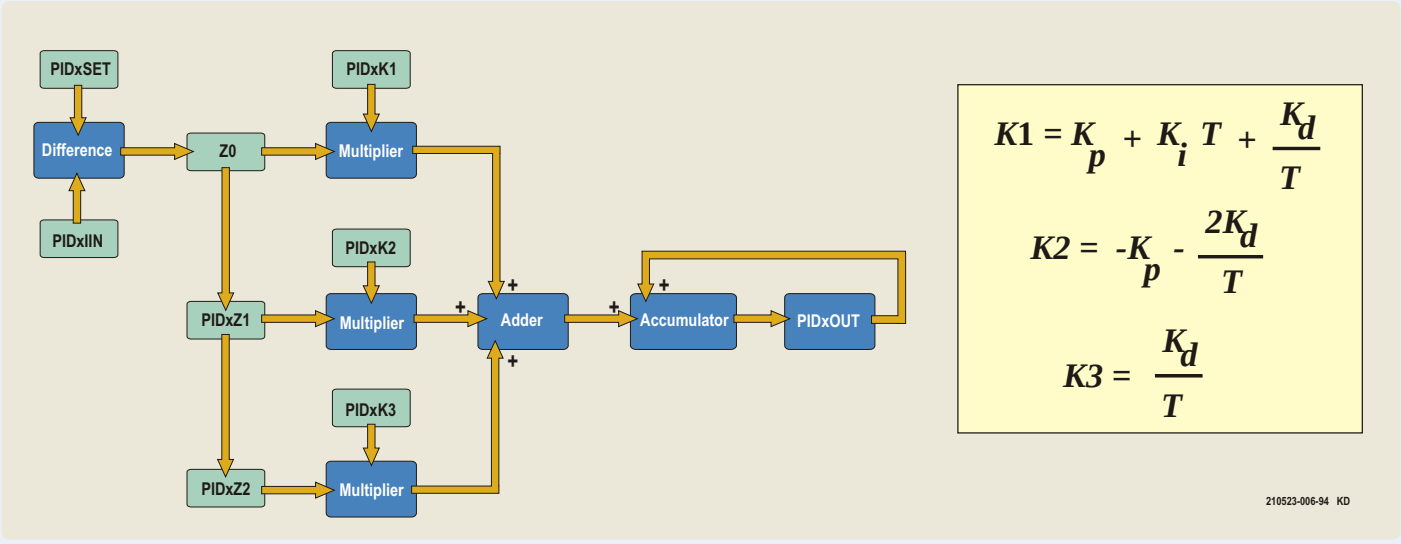


Figure 6. Bloc PID du PIC16F1619 et équations permettant de calculer les trois constantes standard utilisées. (Source : Microchip Technology)

Un moteur BLDC ou PMSM peut être vu comme un CC FHP mécaniquement inversé. Les bobines pilotées en commutation sont enroulées sur le stator (le carter fixe), tandis que l'aimant garnit la paroi du rotor. Le rotor est plus léger avec une faible inertie. La commutation n'est plus mécanique, à charge pour l'ingénieur de concevoir un pilotage électronique. La principale différence entre ces deux types est que les BLDC produisent une FCEM trapézoïdale et sont moins exigeants sur la commutation que les moteurs PMSM qui exigent une commutation sinusoïdale et produisent une FCEM sinusoïdale. Les moteurs PMSM sont également plus silencieux.

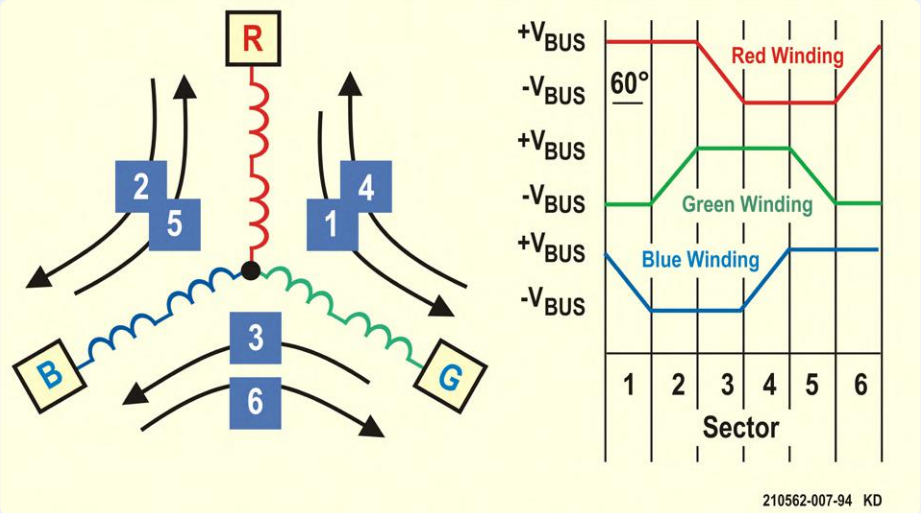


Figure 7. Moteur BLDC avec bobines en étoile et étapes de commutation induisant une rotation électrique élémentaire. La rotation mécanique peut nécessiter plusieurs rotations électriques. (Source : Microchip Technology)

En général, les enroulements sont câblés en étoile. Cela implique six demi-ponts pour alimenter les trois bobines d'un moteur BLDC selon un schéma en six étapes qui applique la puissance à deux des trois (**fig. 7**). Le champ magnétique résultant tourne autour du stator, et produit dans le rotor le couple mécanique souhaité qui l'entraîne. En inversant l'ordre de la séquence de commutation, le rotor change de sens. Bien que ce schéma de principe simple du BLDC rende son fonctionnement clair, en pratique, les bobines sont en général disposées de sorte qu'il faut plusieurs rotations électriques pour obtenir un tour mécanique complet.

Détermination sans capteur de l'angle du rotor

Le défi suivant est de déterminer la position du rotor pour s'assurer que la commutation suivante est lancée au bon moment. Les moteurs BLDC embarquent souvent des capteurs à effet Hall qui donnent l'angle du rotor avec la précision requise. Les fronts montants et descendants fournis déclenchent des interruptions logicielles du MCU, ce qui permet une mise en œuvre simple du pilotage correct des bobines du moteur par des routines d'interruption. Un compteur périodique peut aussi capturer ces signaux pour déterminer la vitesse alors renvoyée sur le contrôleur PID. La vitesse du moteur est également commandée par des signaux MLI. Ainsi, bien que la commutation de ces moteurs repose sur le savoir-faire de l'ingénieur d'étude, l'approche n'est pas si différente de celle des moteurs CC à balais.

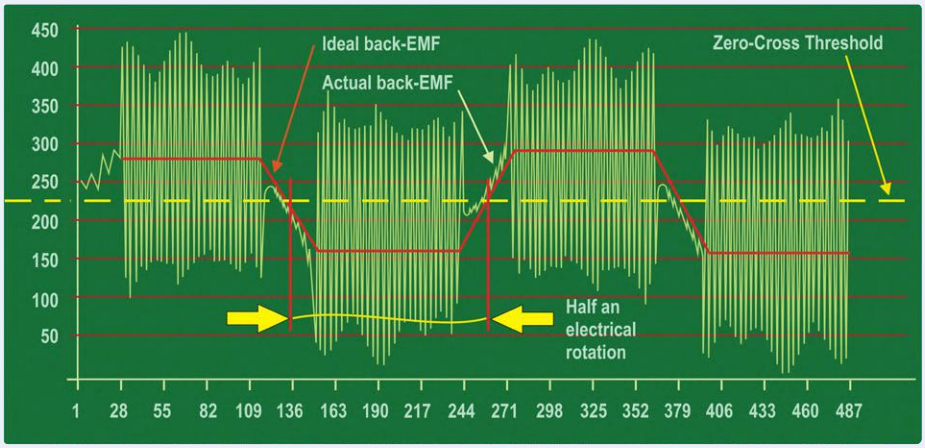


Figure 8. La FCEM de la phase ouverte peut être mesurée et utilisée pour déterminer la rotation angulaire et éviter ainsi l'utilisation d'un capteur à effet Hall. (Source : Microchip Technology)

Mais comme déjà mentionné, les capteurs grèvent le coût et parfois il n'y a tout simplement pas de place pour les accueillir. Comme les moteurs CC à balais, les BLDC produisent une FCEM. Et c'est un fait, dans ces moteurs, ce signal peut servir à déterminer la position du rotor. Lorsque la FCEM de la phase non alimentée passe théoriquement par zéro (à la moitié de la tension alternative d'alimentation), l'angle du rotor peut être déterminé en mesurant la période de ces passages (**fig. 8**).

La théorie semble simple, mais la mise en œuvre reste particulièrement difficile. Tout d'abord, à l'arrêt, il n'y a pas de FCEM, ce qui oblige à faire tourner le moteur à une vitesse telle qu'un algorithme dispose d'un signal de FCEM suffisant pour être mesuré. Ensuite, le code requis pour évaluer les

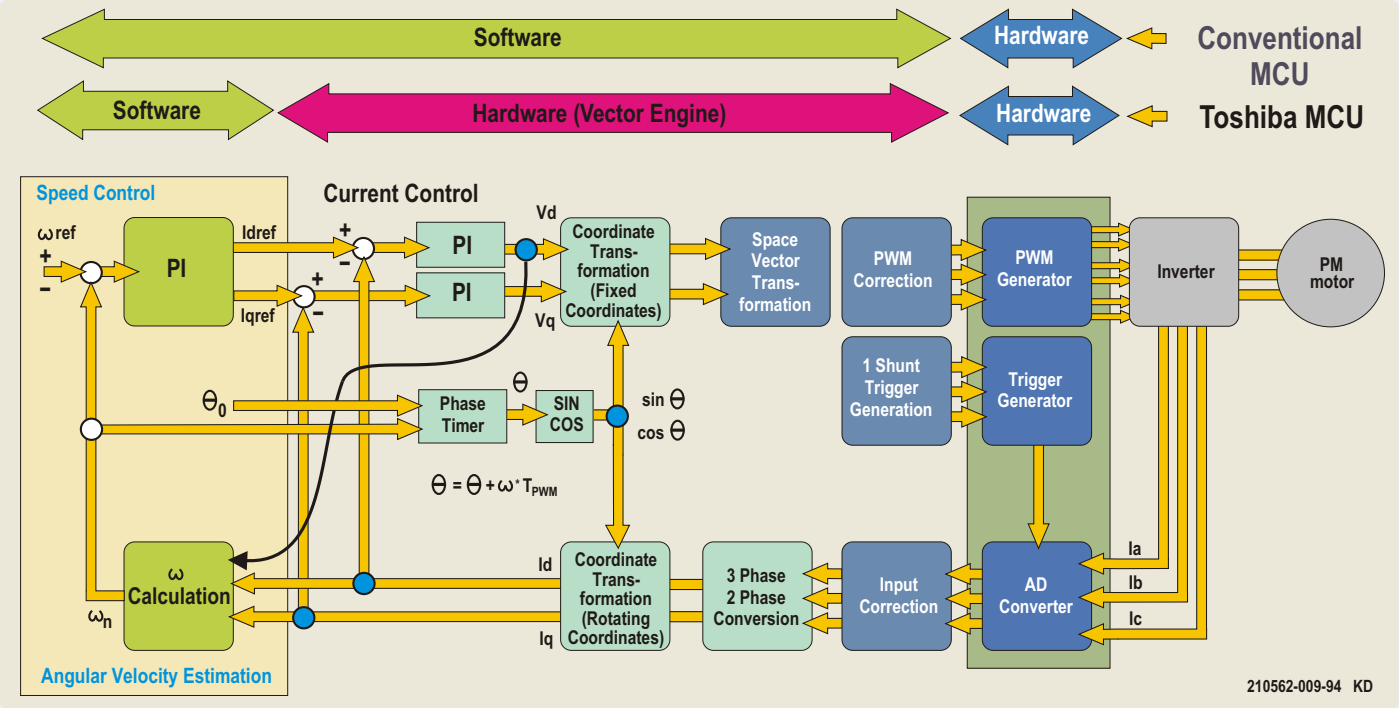


Figure 9. Les modules à haute intégration comme l'Advanced Vector Engine Plus de Toshiba, permettent une commande quasi autonome des moteurs à aimant permanent. (Source : Toshiba Electronics Europe)

données de FCEM et les filtrer peut demander un temps d'exécution long. Cela peut impliquer de passer d'une stratégie de commutation à une autre lorsque le moteur change de régime de fonctionnement. Les fournisseurs de commandes de moteurs BLDC élaborées ont résolu bon nombre de ces problèmes, permettant de caractériser quasi automatiquement un moteur inconnu et d'en faire l'apprentissage en le reliant à une commande.

Mise en œuvre d'une commande sinusoïdale de moteur

L'algorithme de commande PMSM a un niveau supplémentaire de complexité. Il exige que les signaux MLI du MCU soient modulés en continu pour produire une sinusoïde d'alimentation des enroulements du moteur. Une telle commande visant le champ (*Field oriented control* = *FOC*) utilise le courant des enroulements triphasés pour déterminer l'angle instantané du rotor. Exploitant les transformées de *Clarke/Park*, ce procédé de commande tend à maintenir l'orthogonalité des champs magnétiques du rotor et du stator afin d'obtenir un couple maximal à tout moment.

Face à cette complexité, les leaders de la commande de moteurs tentent de simplifier leurs produits afin que les ingénieurs s'efforcent de développer les caractéristiques avantageuses de leurs applications.

Bien qu'une telle complexité puisse être confiée au logiciel sur certains MCU, de nombreux fabricants ont opté pour des modules matériels qui en automatisent une grande partie. Les MCU *TXZ4A+* de *Toshiba* sont un exemple de ces modules [6].

Ces MCU *Arm® Cortex®-M4* comportent un module *Advanced Vector Engine Plus* (*A-VE+*) étroitement couplé aux modules MLI et CA/N (**fig. 9**). Dans les modules A-VE, les transformées de Clarke/Park complètes, de triphasé vers diphasé et inversement, sont mises en œuvre dans le matériel ne nécessitant qu'un régulateur PI, une estimation logicielle de la vitesse angulaire et un peu de gestion interne. En dehors de cela, les modules fonctionnent ensemble en quasi-autonomie.

Innover dans la commande des moteurs

Face à cette complexité, les leaders de la commande de moteurs s'essaient à simplifier leurs solutions afin que les développeurs se concentrent sur les caractéristiques avantageuses de leurs applications. Selon *Jonas P. Proeger*, directeur commercial chez *Trinamic*, aujourd'hui dans le giron d'*Analog Devices* (**fig. 10**), les principaux thèmes abordés concernent l'amélioration de l'efficacité, la réduction du bruit et l'ajout d'intelligence. Une meilleure efficacité est nécessaire pour le nombre croissant de véhicules autonomes

Figure 10. Jonas P. Proeger, directeur de la gestion commerciale chez Trinamic, aujourd'hui filiale d'Analog Devices. (Source : Analog Devices)

sur batterie, par ex. en logistique, les robots autonomes mobiles (AMR). Les moteurs utilisés changent, les densités de puissance plus élevées et les économies de poids se traduisent par des moteurs d'inductance plus faible. En conséquence, les algorithmes et boucles de commande doivent changer et fonctionner jusqu'à 100 kHz. Des CI tels que le *TMC4671* [7] y parviennent en utilisant un matériel spécifique, avec des approches alternatives à base de DSP ou de FPGA.

Le silence de fonctionnement est une autre exigence croissante pour une perception de meilleure qualité que les produits alternatifs à moteurs et réducteurs bruyants. Jonas présente ici sa technique *stealthChop* qui utilise une méthode de hachage contrôlée en tension qui peut réduire le bruit audible de 10 dB pour un moteur pas-à-pas bipolaire [8]. La réduction du courant d'ondulation résiduel diminue l'énergie dissipée par le moteur. Les moteurs intelligents sont le domaine ultime de croissance, là où les industries ont besoin de moteurs « auto-conscients ». Ces moteurs développent une intelligence artificielle (IA) croissante dans les systèmes en fournissant des données en temps réel depuis les ateliers.

Sans contrôle, le pouvoir n'est rien

L'idée que les moteurs BLDC et PMSM deviennent prépondérants est erronée. Il existe toujours de nombreuses applications adaptées aux moteurs CC FHP classiques. Et, avec le grand nombre de ponts en H et MCU d'entrée de gamme dotés de périphériques intelligents, il est clair que l'industrie des semi-conducteurs a encore beaucoup à dire. Toutefois, comme les applications passent en mode batterie, exigent des rendements toujours plus élevés ou un fonctionnement silencieux, les ingénieurs se tourneront vers les moteurs BLDC et PMSM. Bien que beaucoup de techniques de commande des moteurs CC soient toujours exploitées, de nombreux pièges retardent l'utilisation fine et la mise en œuvre des contrôleurs. Une base théorique solide reste utile et permet de déboguer les commandes de moteur. Toutefois, pour gagner du temps, il est utile de s'appuyer sur l'expérience d'experts de ce domaine dynamique et d'envisager les CI à haute intégration du marché.

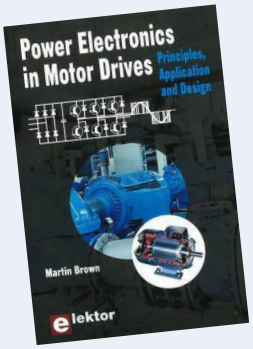
210562-04

Des questions, des commentaires ?

Envoyez un courriel à l'auteur (stuart.cording@elektor.com).

Contributeurs

Conception et texte : **Stuart Cording**
Rédaction : **C. J. Abate**
Illustrations : **Patrick Wielders**
Mise en page : **Harmen Heida**
Traduction : **Yves Georges**



PRODUITS

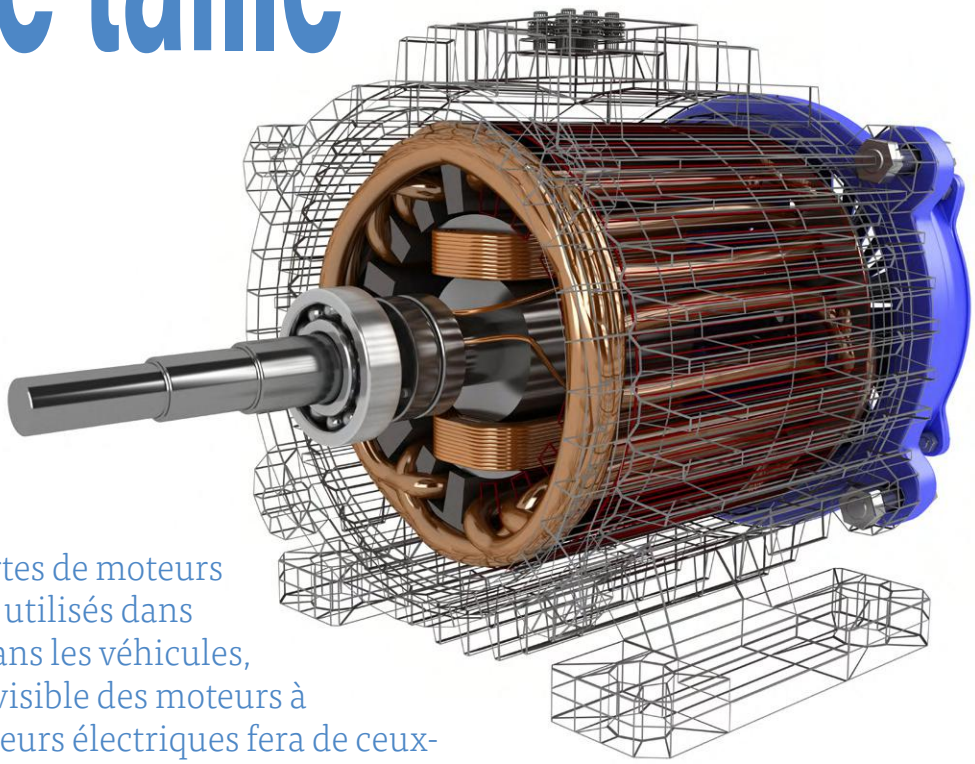
› **E-book en anglais, « Power Electronics in Motor Drives »**
www.elektor.fr/18517

LIENS (tous en anglais)

[1] C. Valens, « How to Choose Between High-side and Low-side Switching » (Comment choisir entre commutation côté Vcc et côté masse), Elektor, 08/2019 : https://bit.ly/3H38uA5.
[2] Vidéo, « Understanding PID Control, Part 1: What Is PID Control? », The MathWorks Inc., 05/2018 : https://bit.ly/3CZfcor
[3] Page du IFX9201SG, Infineon : https://bit.ly/3CZfEmD
[4] Page du kit pont en H, 2GO, Infineon : https://bit.ly/3ww91FZ
[5] D. Hou, « PID Control on PIC16F161X by using a PID Peripheral », Microchip Technology Inc., 2015 : https://bit.ly/3DbGCHy
[6] MCU de la série TXZ4A+, Toshiba : https://bit.ly/3bXUSId
[7] Page du TMC4671, Trinamic : https://bit.ly/3bWO2CA
[8] B. Dwersteg, « Performance stealthChop (qualitative) », TRINAMIC Motion Control GmbH & Co. KG, 02/2018 : https://bit.ly/3CZgxvt

moteurs électriques de grande taille

Principes de base et informations utiles



Motor Control

Thomas Scherer (Elektor)

Depuis des décennies, toutes sortes de moteurs électriques de toutes tailles sont utilisés dans une multitude d'applications. Dans les véhicules, le remplacement généralisé prévisible des moteurs à combustion interne par des moteurs électriques fera de ceux-ci le principal moyen de produire le mouvement mécanique. Cela justifie le fort regain d'intérêt dont ils sont l'objet.

La conversion d'énergie électrique en énergie mécanique est presque exclusivement confiée aux moteurs électriques. Les autres méthodes de production d'énergie mécanique comme les moulins à eau ou à vent et les moteurs thermiques sont loin d'avoir les avantages des moteurs électriques, à savoir faible encombrement, poids réduit, longue durée de vie et rendement élevé. Pour les grandes machines, le rendement peut nettement dépasser 90 %. Les montres non mécaniques de luxe sont mues par de minuscules moteurs. Jusqu'à 100 W de puissance, il y a les moteurs dits *universels* classiques, et les variantes particulières que sont les moteurs pas à pas, servos, moteurs à bague de déphasage, petits moteurs synchrones et à induction et autres moteurs *BLDC* (CC sans balais), de plus en plus répandus. Côté *Elektor*, l'article de *Mathias Claussen* [1] décrit comment piloter les petits moteurs, et mon article d'il y a trois ans [2] traite des techniques de pilotage des moteurs en général. Cet article se concentrera surtout sur les principes applicables aux gros moteurs de plus en plus utilisés en électroménager (par ex. aspirateurs et machines à laver), mais aussi dans les véhicules électriques et, surtout, par l'industrie (par ex. machines-outils).

Types de moteurs et règles empiriques

Comme indiqué ci-dessus, cet article laisse de côté les moteurs de faible puissance, en particulier les plus courants dans ce domaine : à bague de déphasage, à condensateur, pas à pas et servomoteurs. Nous n'étudierons pas non plus les moteurs linéaires, bien qu'ils soient fréquents dans les disques durs (positionnement des têtes) et dans les actionneurs (à une puissance plutôt élevée). Les moteurs linéaires de forte puissance (de l'ordre du MW) ne feront pas partie de notre propos : leurs applications militaires (par ex. dans les canons à rail) pour accélérer des projectiles nécessitent des techniques de pilotage trop spécifiques pour être abordées. Nous nous limiterons plutôt aux quatre types de moteurs du **tableau 1**. Ceux-ci couvrent plus de 90 % des applications pour des puissance comprises entre moins de 1 kW et des centaines de MW. En théorie, chaque type de moteur pourrait être fabriqué en petite ou grande taille, mais en raison de caractéristiques électriques et mécaniques différentes, chacun est adapté à des domaines d'application particuliers, avec des variations considérables entre types. Les ingénieurs non spécialisés se font de fausses idées sur les

Tableau 1. Types de moteurs électriques de grande taille.	
Type	**Caractéristiques**
Moteur universel	Un collecteur mécanique crée un champ alternatif ; nombreuses variantes, à enroulement en série et en dérivation ; couple de démarrage élevé.
Moteur à induction triphasé	Rotor passif (sans bagues collectrices) ; le rotor glisse derrière le champ du stator ; une protection contre les courants d'appel élevés (commutation étoile-triangle ou entraînement à fréquence variable) est nécessaire.
Moteur synchrone triphasé	Champ d'excitation permanent ou électrique requis ; le rotor se déplace de manière synchrone avec le champ tournant ; peut utiliser un entraînement à fréquence variable ; peut fonctionner comme un générateur ; peut être configuré avec des pôles internes ou externes.
Moteur BLDC	Variante du moteur synchrone à commutation électronique ; convient aux trains d'entraînement ; très fiable ; les aimants permanents sont souvent utilisés dans les applications de faible à moyenne puissance.

moteurs électriques, faisons d'abord tomber ces mythes. Prenons par ex. la proposition trop idéalisée selon laquelle, au démarrage (c.-à-d. à un régime de 0 tr/min), les moteurs électriques auraient en théorie un couple « infini ». C'est justement le « en théorie » qui pose un problème : cela supposerait un courant extrême, un champ magnétique extrême, une résistance d'enroulement nulle, des métaux hypersolides et une électronique de commande miraculeuse. En pratique tous ces paramètres trouvent vite leur limite. Le mythe devient une règle d'or selon laquelle la plupart des moteurs électriques peuvent en effet offrir un couple de démarrage élevé si le circuit de commande et l'alimentation sont conçus pour cela. Dans les véhicules électriques, cette caractéristique présente l'avantage, par rapport aux moteurs thermiques, de pouvoir en général se passer de boîte de vitesses manuelle ou automatique. La chaîne cinématique s'en trouve fiabilisée, simplifiée et plus légère : ce dernier aspect est important, même si le gain de poids est plus qu'annihilé par les batteries.

Une autre règle d'or sera utile à ceux qui ne sont pas familiers avec le sujet : à charge donnée, la relation couple/courant de fonctionnement d'un moteur électrique est très linéaire, car le champ magnétique produit est proportionnel au courant qui circule dans les enroulements ; la force magnétique d'attraction ou de répulsion et donc le couple qui en résulte sont proportionnels à ce champ.

Une autre règle d'or, un peu plus grossière, dit que le régime maximal possible à vide (hors considérations mécaniques) dépend de la tension d'alimentation. Pourquoi ? Un moteur électrique en rotation crée ce que l'on appelle une « force contre-électromotrice » ou « FCEM » qui en théorie est bien proportionnelle au régime de rotation. Le terme « force électromotrice » ou « FEM » fut historiquement introduit pour décrire la *tension* produite par un générateur ; par FCEM, nous entendons la tension induite par la rotation d'un moteur et qui tend à s'opposer à la tension d'alimentation. En négligeant les inductances et les pertes, la tension effective présente aux bornes des enroulements du moteur est la *différence* entre la tension appliquée et la FCEM. Le courant résultant dépend presque exclusivement de cette différence et de la résistance ohmique des enroulements. Quand la différence = 0 V, le régime maximal possible est atteint : le petit courant résiduel ne fait que compenser les pertes par frottement.

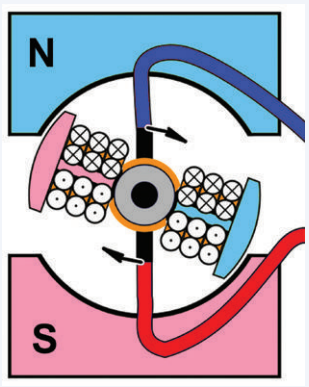


Figure 1. Principe de fonctionnement et construction d'un moteur électrique simple à aimants permanents. Le pôle nord se trouve en haut du stator, le pôle sud en bas. Le rotor est un électroaimant alimenté par un collecteur composé d'une bague segmentée et de balais de carbone. (Crédit dessin : Michael Frey, CC 3.0 [3])

Principes de fonctionnement

La forme de moteur la plus ancienne et, de fait, la plus simple est le moteur à aimant permanent illustré à la **figure 1**. Le rotor est constitué d'un électroaimant monté sur paliers pour pouvoir tourner. Le point noir au milieu représente l'axe de rotation et l'anneau gris un palier. La tension continue appliquée passe par une bague collectrice avec des contacts en cuivre (le collecteur) et des balais en carbone. La polarité de la tension appliquée à l'électroaimant, et donc du champ magnétique, dépend donc de la position du rotor. Si la polarité de la tension appliquée est inversée, le sens de rotation s'inverse également. Il est clair que ce type de moteur constitue un générateur si on entraîne mécaniquement son rotor : un signal continu pulsé apparaît alors sur les balais. En l'absence de collecteur, la machine devient un générateur de courant alternatif ; et en appliquant une tension alternative, on obtient en théorie un moteur synchrone monophasé, mais peu pratique, car il a bien du mal à démarrer.

Presque tous les autres types de moteurs dérivent de ce concept simple. Si l'aimant permanent du stator est remplacé par un enroulement (un autre électro-aimant), le moteur ainsi obtenu est dit *universel*. En introduisant plusieurs phases, nous pouvons

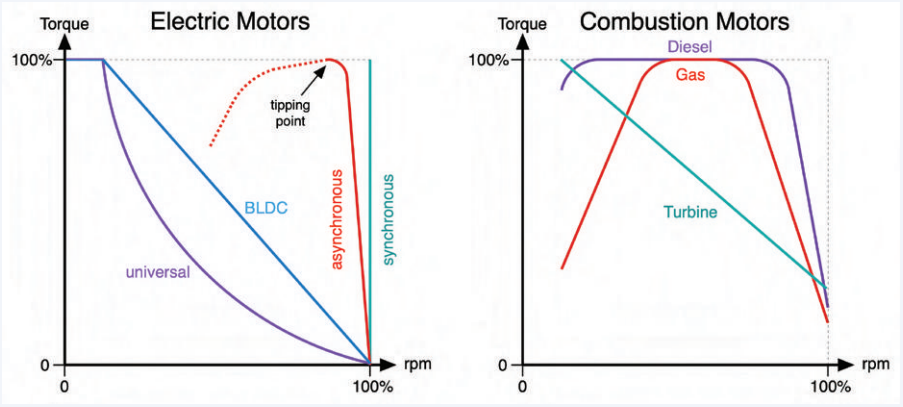


Figure 2. Comportement idéal du couple en fonction du régime pour différents types de moteurs électriques, comparé à trois types de moteurs thermiques. (Crédit dessins : Thomas Scherer)

construire des moteurs à induction, synchrones et BLDC qui utilisent un champ tournant produit électriquement. Le moteur simple a de mauvaises propriétés en roue libre (en l'absence de tension), car les lignes de champ des aimants permanents provoquent toujours un effet de freinage. Sur les moteurs à excitation séparée, ce problème est moindre.

Les caractéristiques principales de ces types de moteurs sont illustrées (dans le cas idéal) et comparées à celles des moteurs thermiques de conception actuelle à la **figure 2**. Les contrastes des courbes couple/régime sont particulièrement remarquables. Cette comparaison est loin d'être exhaustive, et chaque caractéristique a une influence notable sur l'adéquation de chaque type de moteur à différents scénarios d'application.

Le moteur universel

La **figure 3** montre le moteur universel, inventé il y a plus de 117 ans. C'est la variante la plus proche du moteur simple de la fig. 1. Ici, un 2e enroulement constitue un électroaimant qui remplace simplement l'aimant permanent. Les avantages du moteur universel sont patents, surtout dans le domaine des hautes puissances, où les aimants permanents requis deviennent non seulement très grands et lourds, mais aussi coûteux. Dans le moteur universel classique, l'enroulement d'excitation du stator est câblé en série avec celui du rotor, formant le moteur dit à enroulements série. Le principal avantage de ce moteur par rapport au moteur à enroulements shunt (non illustré), moins répandu, où l'enroulement du stator est câblé en parallèle avec l'enroulement du rotor, est qu'il a un couple de démarrage très élevé.

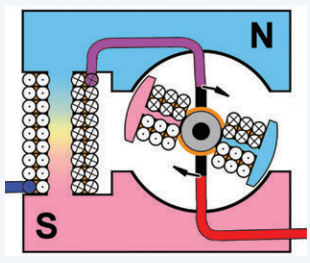


Figure 3. Principe de fonctionnement et construction d'un moteur universel, où l'aimant permanent d'un moteur simple est remplacé par un électroaimant. (Modifications de la figure 1 : Thomas Scherer)

Les deux variantes peuvent être alimentées en courant continu ou alternatif. Les modèles optimisés pour l'alternatif avec haut régime utilisent, pour le noyau du stator et du rotor, un empilement de feuilles métalliques isolées entre elles qui réduit les pertes par courant de Foucault et hystérésis. Avec les deux variantes, l'inversion de polarité de la tension d'alimentation en marche n'affecte pas le sens de rotation. Pour inverser le sens, il est nécessaire de changer la polarité du stator *ou* du rotor (et non des deux simultanément).

Les gros moteurs universels (de l'ordre du kW) ont un courant de démarrage élevé. C'est pourquoi ils utilisent pour se protéger (et empêcher leurs fusibles de sauter), un contrôleur de démarrage. Le plus simple alimente initialement le moteur via une résistance série qui réduit le courant d'appel. Après un court laps de temps, ou dès qu'un régime donné est atteint, la résistance est shuntée. Sa valeur doit être choisie en fonction des conditions de fonctionnement particulières et elle doit être capable de dissiper la chaleur produite, soit l'intégrale de la courbe de puissance pendant la période de démarrage. En définitive, la résistance de démarrage requise peut être assez encombrante, nécessiter une construction spéciale, ou même prendre la forme d'un rhéostat liquide. Pour un très gros moteur universel, une seule résistance ne suffit pas : on utilise plusieurs résistances en série. On les court-circuite une à une au fur et mesure de la montée en régime. Le plus simple contrôle de protection au démarrage est la temporisation illustrée à la **figure 4.** La mesure du régime couplée à la mise en court-circuit échelonnée des résistances, le cas échéant avec une augmentation progressive de la tension d'alimentation par un contrôleur électronique, est une approche plus intelligente.

Comme mentionné ci-dessus, les moteurs universels peuvent être alimentés par une source en CA ou CC. Le régime en charge et à vide dépend de la construction du moteur et de la tension appliquée. La puissance de sortie et le régime en charge peuvent donc être contrôlés à l'aide d'une commande MLI. Si un tel circuit de commande MLI est déjà disponible, il est facile de l'associer au microcontrôleur existant pour fournir une protection au démarrage. Pour ce faire, il suffit de démarrer le fonctionnement à l'aide d'un signal

MLI avec un rapport cyclique élevé, puis de le réduire progressivement. Si le moteur se bloque à cause d'une charge excessive, ou même en cas de simple surcharge, des courants potentiellement très élevés et destructeurs circulent. Il est donc recommandé de surveiller non seulement le régime du moteur, mais aussi le courant d'entraînement, surtout pour les gros moteurs. Dans les cas simples, on peut se contenter d'un coupe-circuit thermique de surintensité. Si le régime tombe en deçà du seuil minimal ou si le courant dépasse une limite supérieure, le moteur est mis hors tension, protégeant le moteur lui-même, son électronique de commande et toute machine entraînée. Par rapport à un fusible, cette protection présente l'avantage de ne pas interrompre l'alimentation d'autres machines alimentées par les mêmes lignes.

Comme leur nom le suggère, les moteurs universels sont dans toutes sortes d'applications : des appareils électroménagers, perceuses et autres outils d'une puissance inférieure à 100 W jusqu'aux locomotives électriques et autres unités motrices de plusieurs MW, alimentées en CC ou CA.

La variante simple présentée à la figure 3 souffre d'un couple irrégulier en fonction de la position de rotation. Les schémas plus sophistiqués sont donc la règle. Les systèmes monophasés multipolaires avec une variété de dessins de stator et de rotor (voir **fig. 5**) sont courants avec divers types d'enroulements de compensation qui touchent la courbe du couple, mais aussi la quantité d'étincelles sur les balais. La fiabilité du collecteur et d'autres propriétés sont aussi touchées. Si l'on examine la relation entre le couple et le régime sur la figure 2, on constate qu'à une tension donnée, le régime varie en fonction de la charge. Pour stabiliser le régime, il faut une boucle de contrôle active et un capteur de régime.

Le moteur à induction

Le moteur à induction, dont la construction de base est illustrée à la **figure 6**, présente des avantages décisifs par rapport au moteur universel. Le plus important d'entre eux est l'absence totale de collecteur : l'absence d'usure (hormis des paliers) assure une très bonne fiabilité. Les plus gros moteurs à induction sont presque toujours triphasés, et les modèles multipolaires de ce type fournissent un couple régulier sur chaque rotation. Le dessin courant à rotor interne, tel qu'illustré, utilise le champ magnétique rotatif du stator : en charge, le rotor tourne un peu plus lentement que le champ du stator, induisant un courant dans ses enroulements et donc un champ magnétique. La FCEM correspondante induit à son tour un courant accru dans l'enroulement du rotor, d'où le terme « moteur à induction ». La **figure 7** montre un moteur à induction dont la puissance nominale est de quelques kW. Le nombre de pôles choisi détermine le régime nominal du moteur : doubler le nombre de pôles divise par deux le régime moteur à fréquence égale de l'alimentation triphasée.

Comme on le voit sur le graphique de la figure 2, la courbe de couple du moteur à induction est très raide et le rotor tourne toujours à un régime juste inférieur à la vitesse de rotation du champ appliqué. C'est ce qu'on appelle le « glissement ». Son importance dépend de la charge, mais il reste étroitement lié à la vitesse de rotation du champ (et donc à la fréquence de l'alimentation CA). Lorsque la charge du moteur augmente, un seuil est finalement atteint, après quoi le couple chute brusquement. Le moteur cale alors et

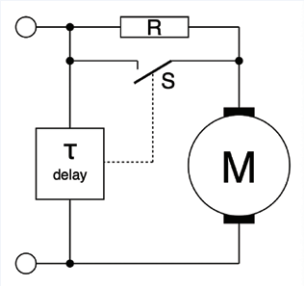


Figure 4. Circuit simple de protection au démarrage utilisant une résistance en série pour limiter le courant d'appel.

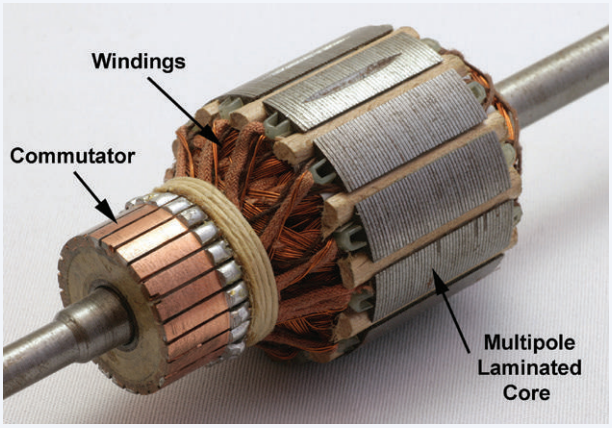


Figure 5. Rotor d'un moteur universel monophasé type. Le nombre de contacts sur le collecteur correspond aux nombres d'enroulements et de pôles. (Crédit photo : Sebastian Stabinger, CC 3.0 [4])

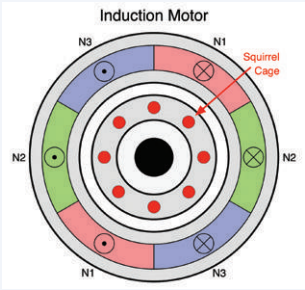


Figure 6. Schéma de construction d'un moteur à induction triphasé (rotor interne). Le rotor est constitué d'une « cage d'écureuil » à enroulements court-circuités. (Crédit dessin : Thomas Scherer)

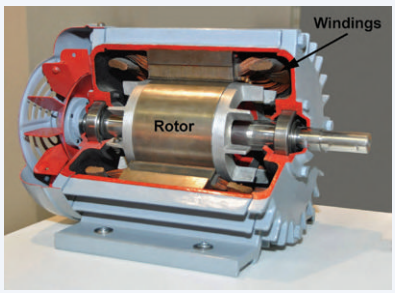


Figure 7. Moteur à induction de quelques kW de puissance nominale (coupe partielle). On distingue facilement les enroulements du stator et le rotor en court-circuit. (Crédit photo : S. J. de Waard, CC 3.0 [5])

Figure 8. Petit variateur de fréquence pour moteurs à induction de quelques kW. Ce circuit de commutation électronique permet de régler le régime. Des modèles plus conséquents offrant des fonctions avancées sont disponibles. (Crédit photo : C. J. Cowie, CC 3.0 [6])

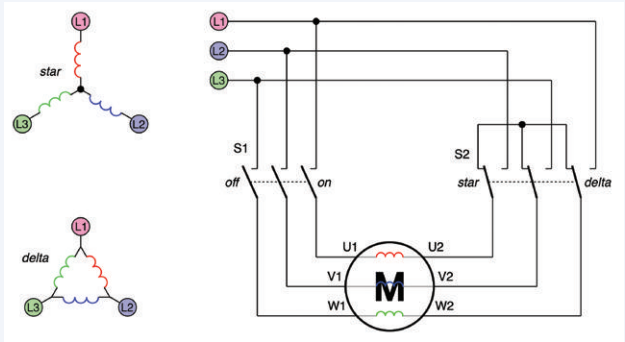


Figure 9. Commutation étoile-triangle de protection. C'est un moyen simple de réduire les forts appels de courant au démarrage du moteur.

le courant augmente rapidement. La plage de régimes utilisables peut être considérablement ajustée en alimentant le moteur par un variateur de fréquence (voir **fig. 8**). Ce type d'entraînement est très répandu sur les machines industrielles.
Les gros moteurs à induction nécessitent un courant élevé au démarrage : jusqu'à dix fois le courant nominal de fonctionnement.

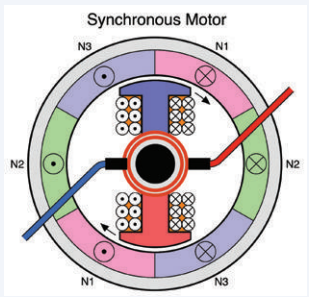


Figure 10. Schéma de construction d'un moteur synchrone triphasé simple. Le rotor peut être équipé d'aimants permanents pour éliminer le collecteur. (Crédit dessin : Thomas Scherer)

Il existe plusieurs façons de réduire le courant d'appel. Dans les cas simples, disons jusqu'à 11 kW, on utilise un commutateur étoile-triangle (**fig. 9**) : la configuration en étoile absorbe un courant et (donc) une puissance égaux au tiers de ceux de la configuration en triangle. Le passage étoile-triangle de la configuration s'effectue généralement lorsque le moteur a atteint au moins 75 % de son régime nominal. Il faut veiller à ce que la charge ne soit pas trop élevée pendant le démarrage afin de rester en deçà du couple de décrochage. Avec un variateur, cette précaution n'est plus nécessaire, car la fréquence du variateur et donc le régime peuvent être augmentés progressivement au démarrage.
En raison de leur extrême robustesse et de leur haut rendement, on trouve ce type de moteur dans toutes sortes de machines en production industrielle ainsi que dans les chaînes cinématiques des véhicules électriques. Sur ses modèles actuels, Tesla utilise de tels moteurs, compacts, tournant à haut régime et couplés à un réducteur. Un système intelligent de contrôle du moteur combinant des circuits de surveillance du courant et du régime couvre la très vaste gamme de régimes requise. La commande électronique se charge de supprimer le principal inconvénient du moteur, à savoir son lien étroit entre régime et fréquence d'alimentation en CA. Si le rotor tourne plus vite que le champ tournant du stator, le moteur à induction fonctionne comme un générateur : cela permet de récupérer de l'énergie, ce qui est important pour les véhicules comme pour les machines fixes.

Moteurs synchrones et BLDC

Un moteur synchrone est essentiellement un moteur universel fonctionnant sur une alimentation triphasée. Selon un schéma répandu, l'enroulement du rotor est alimenté par des bagues collectrices séparées : un collecteur de type moteur universel n'est pas nécessaire et les bagues ne sont pas segmentées, réduisant les frottements. La **figure 10** montre un modèle de ce type ; le rotor peut être à aimants permanents, rendant les bagues totalement inutiles. Pour les gros moteurs fixes de grande taille, l'excitation séparée illustrée est plus économique et plus fiable d'un point de vue thermique : même brève, une exposition à une température élevée peut facilement endommager les aimants permanents. Il existe également des modèles à rotor externe, où le stator est interne et le rotor est équipé d'aimants permanents, donc sans bagues collectrices. Ces types ont une longue durée de vie et ne nécessitent aucun entretien. Comme on peut le voir sur le graphique du couple en fonction du régime, la vitesse reste constante et est rigidement couplée à la vitesse du champ tournant. Le rotor ne glisse pas sous la charge ; par rapport au champ tournant, il est décalé d'un angle de phase dont la valeur dépend de la charge. Comme les autres types, ce moteur peut être utilisé comme générateur. Les moteurs synchrones sont fabriqués avec des pôles multiples pour régulariser le couple sur une révolution ; ici aussi, le régime est inversement proportionnel au nombre de pôles.
Le grand avantage du moteur synchrone est indiqué par son nom : il permet d'obtenir un régime très précis, constant et indépendant de la charge, sans régulateur. Contrairement au moteur universel, les balais de charbon ne subissent que très peu d'usure, car les bagues collectrices ne sont pas segmentées et l'énergie d'excitation

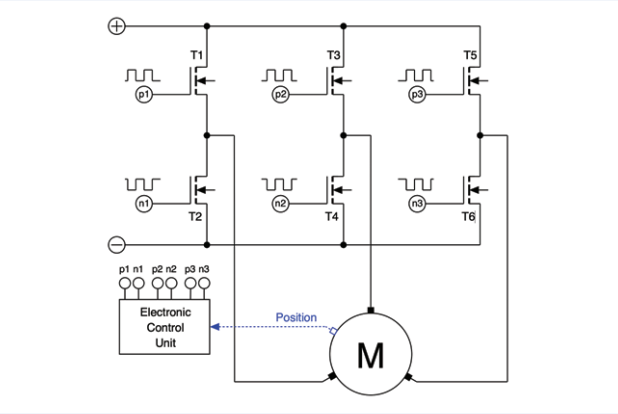


Figure 11. Principe de l'électronique de commutation sur un moteur BLDC. Chaque demi-pont MOSFET reçoit une paire de signaux de commande complémentaires avec un retard de phase approprié.

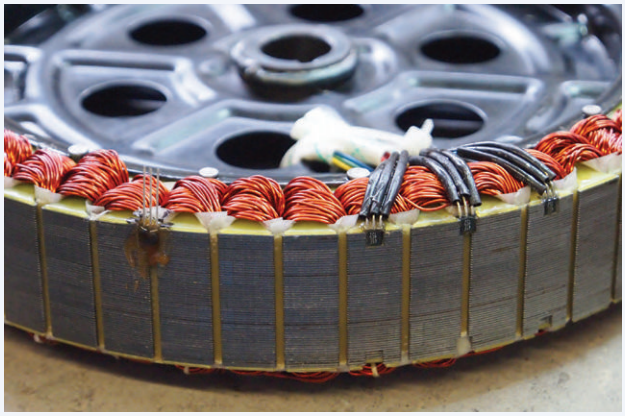


Figure 12. Stator d'un moteur BLDC de 2 kW pour moyeu de roue avec rotor externe. Les enroulements quadripolaires et les tôles sont facilement visibles, tout comme l'emplacement des capteurs à effet Hall utilisés pour détecter la position angulaire du rotor. (Crédit photo : Thomas Scherer)

est relativement faible. Si le rotor utilise des aimants permanents, il est possible de construire des moteurs extrêmement robustes et de petite taille. Un inconvénient est une relative difficulté à démarrer le moteur : pour cette raison, un gros moteur synchrone comprend parfois sur le rotor un bobinage supplémentaire court-circuité, et peut ainsi être mis en vitesse comme un moteur à induction. Au démarrage, les techniques de protection (dont les fonctions de protection offertes par les variateurs de fréquence modernes) déjà utilisées sur le moteur à induction peuvent (parfois, doivent) être mises en œuvre.
Le champ tournant des enroulements du stator d'un moteur synchrone peut être produit en alimentant les enroulements à partir d'une source de courant continu à l'aide de trois demi-ponts, plutôt qu'à courant alternatif triphasé (voir **fig. 11)**. Les demi-ponts sont contrôlés électroniquement pour créer la forme d'onde triphasée, comme dans un variateur de fréquence. Un tel moteur est dit à courant continu sans balais (ou BLDC =*brushless direct current*). Un capteur détecte la position angulaire du rotor, et cette information permet de synchroniser la commande du demi-pont. On obtient ainsi un moteur quasi synchrone qui crée de manière autonome son propre champ tournant et dont le régime dépend de la tension d'alimentation et de la charge. Contrairement au moteur universel, la construction est simple et, si des aimants permanents sont utilisés, le moteur ne nécessite pratiquement aucun entretien. La caractéristique couple-vitesse est également supérieure. Si un contrôleur MLI est ajouté, le régime à vide peut être ajusté en modifiant le rapport d'espacement des signaux de commande. Si le signal MLI est régulé en utilisant le courant d'alimentation comme signal de référence, la relation courant/couple sera très linéaire à charge et vitesse données.
Les machines fixes utilisent les deux types de moteurs sur une large gamme de puissance. En général, les moteurs BLDC sont mieux adaptés à la propulsion des véhicules, car ils permettent de concevoir des solutions tout électroniques avec un contrôle sophistiqué. Le « moteur de moyeu de roue » (voir **fig. 12**) utilisé dans les bicyclettes électriques et d'autres applications est une variante de puissance faible à moyenne. Ces moteurs comportent un rotor externe, généralement équipé d'aimants permanents. Avec un grand nombre de pôles, le régime est faible et le couple élevé. Le couple est indépendant de la position angulaire et la fiabilité est excellente. Cependant, pour les applications de forte puissance, les aimants permanents nécessaires sont à la fois chers et lourds. Le moteur de moyeu de roue constitue donc une masse non suspendue considérable qui nuit au confort de conduite. À cause de cela, les véhicules électriques utilisent le plus souvent des moteurs à induction rapides et compacts entraînant les roues par des accouplements flexibles.

De plus…

Les moteurs électriques sont largement utilisés et semblent devoir supplanter tous les autres types de moteurs, mais leur utilisation nécessite de tenir compte de quelques points. Les moteurs électriques sont aujourd'hui très raffinés et des innovations importantes semblent désormais peu probables. Cependant, les optimisations évolutives n'ont pas fini de voir le jour : elles viendront de l'amélioration des matériaux existants et des technologies de construction et de fabrication, mais aussi, et surtout, du contrôle électronique intelligent. Les semi-conducteurs de puissance ne cessent de s'améliorer en vitesse et en efficacité ; le coût de la puissance de calcul ne cesse de baisser. Cela signifie que des algorithmes complexes peuvent être mis en œuvre. Ils améliorent l'efficacité de la combinaison moteur électromécanique et système de contrôle électronique.
En fin de compte, le meilleur compromis entre un grand nombre de facteurs préside toujours au choix d'un moteur électrique pour une application particulière. Ce choix exige un vaste savoir-faire, et les clients se tournent vers les fabricants de moteurs électriques pour obtenir une assistance de qualité et des conseils spécialisés.

Tableau 2. Les vingt premiers fabricants de moteurs électriques (en 2019) [8].

Classement	Fabricant	Pays
1	Siemens	Allemagne
2	Toshiba	Japon
3	ABB	Suisse & Suède
4	Nidec Motor	Japon
5	Rockwell Automation	États-Unis
6	Ametek	États-Unis
7	Regal Beloit	États-Unis
8	Johnson Electric	Hong Kong
9	Franklin Electric	Inde
10	Allied Motion	États-Unis
11	Faulhaber	Allemagne
12	General Electric	États-Unis
13	Danaher Motion	États-Unis
14	CEW	Brésil
15	maxon motor	Suisse
16	TECO Westinghouse	Taïwan
17	Hitachi	Japon
18	Lincoln Electric	États-Unis
19	Piela Electric	États-Unis
20	Dumore Corporation	États-Unis

On compte des centaines de fabricants de moteurs électriques, il est difficile d'avoir une vue d'ensemble du marché. En 2020, le CA total de tous les fabricants de moteurs électriques avoisinait les 150 G$ [7], et la croissance attendue pour les prochaines années est de 6 %/an. Le **tableau 2** énumère les vingt plus grands fabricants : les entreprises basées aux États-Unis dominent, mais le leader du marché est allemand. Si vous cherchez quelque chose sortant des sentiers battus, une simple connaissance des principes de base ne suffira pas : soyez prêt à une immersion de longue durée dans les finesses du sujet !

210555-04

Contributeurs

Auteur : **Thomas Scherer**
Rédaction : **Jens Nickel**
Mise en page : **Giel Dols**
Traduction : **Yves Georges**

Des questions, des commentaires ?

Contactez Elektor (redaction@elektor.fr).

PRODUITS

- **SmartPi 2 - Compteur intelligent pour Raspberry Pi (incluant trois sondes de courant)**
  www.elektor.fr/18165
- **Sonde de courant pour SmartPi 2**
  www.elektor.fr/18167
- **Pince de mesure PeakTech 4350**
  www.elektor.fr/18161

LIENS

[1] M. Claussen, « Commande de moteur avec des ponts en H », Elektor 01-02/2022 : https://www.elektormagazine.fr/210491-04
[2] T. Scherer, « La commande des moteurs électriques », Elektor 01-02/2019 : https://www.elektormagazine.fr/180558-04
[3] Moteur électrique simple : https://commons.wikimedia.org/wiki/File:Animation_einer_Gleichstrommaschine.gif
[4] Rotor d'un moteur universel : https://commons.wikimedia.org/wiki/File:Kommutator_universalmotor_stab.jpg
[5] Moteur à induction en coupe : https://commons.wikimedia.org/wiki/File:Rotterdam_Ahoy_Europort_2011_(14).JPG
[6] Variateur de fréquence : https://commons.wikimedia.org/wiki/File:Small_variable-frequency_drive.jpg
[7] Le marché des moteurs électriques : https://www.grandviewresearch.com/industry-analysis/electric-motor-market
[8] Classement des fabricants de moteurs électriques : https://blog.technavio.com/blog/top-20-electric-motor-companies

prise en main du microcontrôleur ESP32-C3 RISC-V

Mathias Claußen (Elektor)

L'ESP32-C3 d'Espressif est arrivé. Alternative économique et monocœur à l'ESP8266, l'ESP32-C3 utilise l'architecture ouverte du jeu d'instructions RISC-V. Faisons le point…

Figure 1. Carte ESP32-C3 DevKitC-02.

L'ESP32-C3 d'Espressif était très attendu. Son fonctionnement n'est cadencé que par un seul cœur, contre deux habituellement pour les contrôleurs basés sur le SoC ESP32. Ce cœur utilise l'architecture libre et ouverte du jeu d'instructions RISC-V, en concurrence avec les contrôleurs à base d'ARM, largement utilisés pour les applications IoT. J'ai récemment jeté un coup d'œil à l'ESP32-C3 et à l'ESP32-C3-DevKitC-02 pour voir comment ce microcontrôleur se comportait en laboratoire. Passons en revue ce que j'ai pu découvrir.

Ceux qui me suivent sur Twitter (@ElektorMathias) [1] ont peut-être remarqué la présence de deux DevKits ESP32-C3 sur mon bureau. Les visiteurs qui ont consulté nos notes de laboratoire du mois de juin [2] savent également que la prise en charge logicielle de l'ESP32-C3 est toujours en cours. Le kit de développement objet de cet article utilise la révision 2 de la puce ESP32-C3. Ces versions préliminaires sont accompagnées d'une feuille d'errata A4 (imprimée recto verso) énumérant les problèmes identifiés à ce jour.

Entre autres, la révision 2 souffre d'« insomnie », c'est-à-dire qu'elle consomme beaucoup d'énergie en mode veille profonde, et l'adaptateur série USB/JTAG intégré à la puce ne fonctionne pas. La version plus récente de la révision 3 devrait avoir résolu ces problèmes, comme je l'ai noté sur Twitter [3].

Le référentiel ESP-IDF d'Espressif associé à l'ESP32-C3 est également en cours de mise en place et peut contenir des bogues. Étant donné que la prise en charge d'Arduino est basée sur cet environnement, les bogues éventuels migreront également sur cette plateforme. Au moment où j'écris cet article, la version de développement du package de prise en charge d'Arduino devra être installée dans l'EDI Arduino que vous utilisez. Pour ce faire, entrez le lien suivant sous *Préférences -> URL de gestionnaire de cartes supplémentaires :* `https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package_esp32_dev_index.json`, ce qui permet d'installer la version 2.0.0-rc1.

Les acheteurs de kits de développement basés sur l'ESP32-C3 devraient recevoir la dernière version (rev.3) de la puce, comme on peut le lire sur le forum d'ESP32.com [4]. La version préliminaire rev.2 problématique n'a été installée que sur un nombre limité des premiers kits de développement fournis.

Carte ESP32-C3 DevKitC-02

Tout d'abord, intéressons-nous à la puce ESP32-C3 et à la carte DevKitC-02 (**fig. 1**). La puce ESP32-C3 succède effectivement à l'ESP8266 d'Espressif. Comme l'ESP8266, elle utilise un processeur monocœur qui peut être cadencé jusqu'à 160 MHz et possède une puce de communication Wi-Fi BGN à 2,4 GHz avec mise en œuvre complète de la pile TCP/IP. Mais les similitudes avec l'ESP8266 s'arrêtent là. L'ESP32-C3 comprend beaucoup plus de périphériques associés à

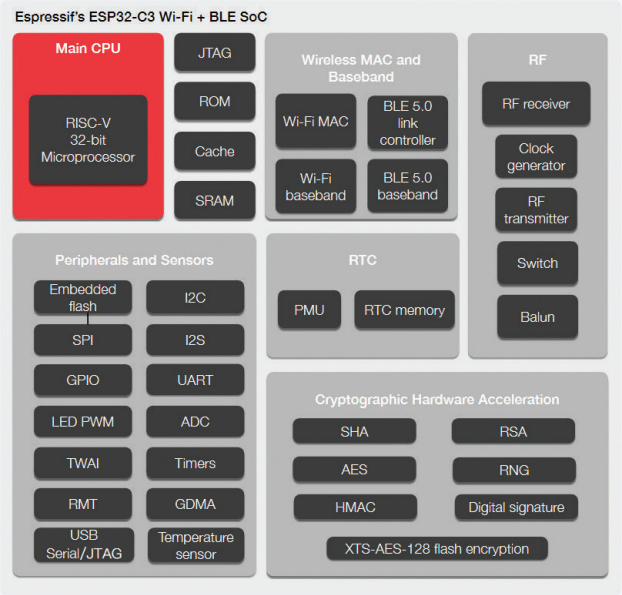


Figure 2. Schéma de principe de la carte ESP32-C3.

- CPU RISC-V cadencée à 160 MHz
- 400 Ko de SRAM (16 Ko de cache Flash)
- Wi-Fi (BGN) 2,4 GHz intégré
- Bluetooth LE 5.0
- Accélérateur matériel de chiffrage
- 22 GPIO programmables
- 2× CA/N SAR à 12 bits
- 3× SPI (supporte SPI, Dual SPI, Quad SPI et QPI)
- 2×x UART (supporte RS232, RS485 IrDA jusqu'à 5 MBd)
- 1× I²C (jusqu'à 800 kbit/s)
- 1× I²S RMT (commande à distance de périphériques)
- TWAI (compatible CAN 2.0 b / ISO 11898-1)
- PWM
- Adaptateur interne USB/JTAG

Tableau 1. Spécifications de l'ESP32-C3.

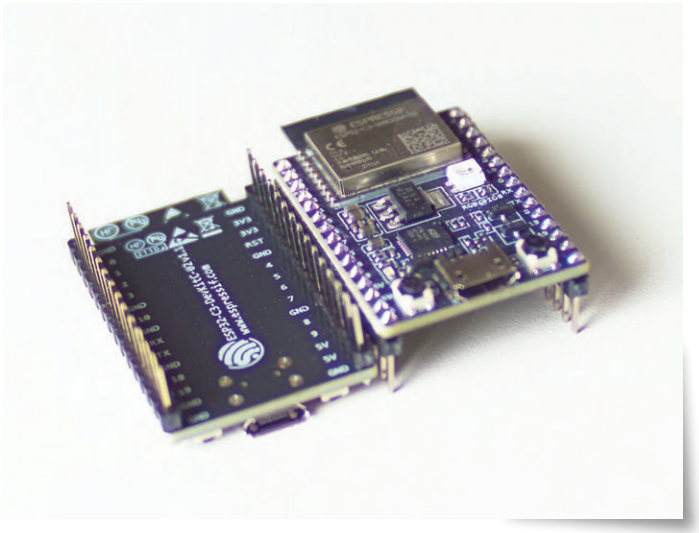


Figure 3. Les deux faces de la carte ESP32-C3-DevKitC-02.

l'ESP32, qui est le successeur de l'ESP8266 chez Espressif. En plus du Wi-Fi, l'ESP32-C3 inclut les communications BLE 5.0 et Bluetooth Mesh. Il utilise également la matrice GPIO particulièrement efficace, de sorte que presque toutes les fonctions peuvent être assignées à quasiment toutes les broches. La **figure 2** présente le schéma fonctionnel de l'ESP32-C3, qui comprend un adaptateur USB série/JTAG.

Un schéma de principe des fonctions du contrôleur est représenté sur la **figure 2** et montre clairement ce qu'il doit à l'ESP32, avec un aperçu des spécifications dans le **tableau 1**. Avec ses 384 Ko de mémoire RAM, l'ESP32-C3 offre presque cinq fois plus d'espace mémoire que l'ESP8266 (80 Ko). Le principal composant qui distingue l'ESP32-C3 de toutes les autres puces ESP32 ou ESP8266 est le cœur du processeur. Alors que ses prédécesseurs utilisent le processeur RISC Tensilica L106 ou LX6/LX7 [5], l'ESP32-C3 dispose d'une unité centrale RISC-V. Cela signifie que les compilateurs et autres programmes de la chaîne d'outils RISC-V peuvent être utilisés avec ce noyau. Les améliorations apportées à ces compilateurs et aux outils correspondants profiteront donc aux utilisateurs de l'ESP32-C3. Il n'est pas nécessaire que la communauté se lance dans la création d'une chaîne d'outils, comme ce fut le cas lors de l'introduction de l'ESP8266.

La carte DevKitC-02 comprend un convertisseur USB-série ainsi qu'une LED RVB compatible WS2812. Un schéma détaillé du circuit de la carte figure [6]. La **figure 3** montre les deux faces de la carte.

Autre routine de clignotement

La LED RVB compatible WS2812 installée sur la carte est commandée à l'aide d'un protocole de communication série. Nous ne pourrons donc pas nous en sortir simplement en configurant une broche d'E/S pour allumer la LED. Fort heureusement, la bibliothèque NeoPixel d'Adafruit contient des routines qui fonctionnent avec ce type de LED. Si elle n'est pas déjà installée, vous pouvez l'ajouter à votre EDI Arduino de la manière habituelle. Le code du **listage 1** fait clignoter la LED RVB en rouge. Vous n'avez pas besoin d'apporter de modifications particulières au code pour qu'il fonctionne sur l'ESP32-C3. Le téléchargement du code est tout aussi facile qu'avec un ESP32. Grâce à la matrice d'E/S, les LED WS2812 peuvent également être utilisées avec d'autres broches.

Lecture du type et de la version de la puce

Il est possible de lire les caractéristiques stockées dans les puces ESP32 elles-mêmes. Le **listage 2** affiche les informations lues via l'interface série à 115200 bauds (pour un débit à 9600 bauds, avec la révision 2 de la puce et le convertisseur USB-série intégré, les caractères sont brouillés). Vous pourrez y lire la version de la puce qui équipe votre kit ESP32-C3. Lorsque vous verrez la liste des bogues connus associés à la révision 2, vous serez heureux de pouvoir confirmer que la version de la puce installée n'est pas affectée.

Portage de projets ESP32

Parmi les caractéristiques les plus utiles de l'ESP32-C3 figurent ses capacités de communication Wi-Fi et BLE intégrées. Associées aux systèmes de fichiers SPIFFS ou LittleFS pour gérer les pages web et d'autres données sur l'ESP32, ces fonctions de communication font de cette plateforme une solution idéale pour un large éventail d'applications Wi-Fi. Ceux qui travaillent avec une ancienne version 1.X de l'EDI Arduino devront installer un *plug-in* corrigé [7] pour télécharger des fichiers sur le système de fichiers de l'ESP32. La version originale de *me-no-dev* [8] ne fonctionne pas avec l'ESP32-C3.

Listage 1. Code du clignotement.

```
/* This sample needs the Adafruit NeoPixel library */
#include <Adafruit_NeoPixel.h>
 #define PIN        8
 #define NUMPIXELS  1
Adafruit_NeoPixel pixels(NUMPIXELS, PIN, NEO_GRB + NEO_KHZ800);

void setup() {
     pixels.begin(); // INITIALIZE NeoPixel strip object (REQUIRED)
     pixels.clear(); // Set all pixel colors to 'off'
 }

void loop() {
     delay(1000);                          // wait for a second
         pixels.setPixelColor(0, pixels.Color(255, 0, 0));
         pixels.show();
     delay(1000);                          // wait for a second
         pixels.setPixelColor(0, pixels.Color(0, 0, 0));
         pixels.show();
 }
```

À titre d'exemple, nous avons utilisé le code destiné à construire un serveur mini-NTP ESP32 [9] [10]. Il peut fournir l'heure via NTP dans son propre réseau et est écrit pour le *framework* Arduino. Après quelques ajustements dans l'affectation des broches, le code peut être compilé sans problème et téléchargé sur l'ESP32-C3. Cet exemple élémentaire montre qu'une grande partie du code existant et des connaissances de l'ESP32 sera directement portable sur l'ESP32-C3, de sorte que même les débutants en matière d'environnements RISC-V ne devraient pas trouver la transition trop ardue. La plupart des exemples ESP32 existants seront également opérationnels sur l'ESP32-C3, et comme FreeRTOS fonctionne en arrière-plan ici aussi, vous retrouverez tous ses avantages et inconvénients, comme avec un ESP32.

Comme l'ESP32-C3 n'est actuellement compatible qu'avec une branche de développement des puces ESP32 dans le *framework* Arduino, tous les EDI ne prennent pas encore pleinement en charge la puce. Au fur et à mesure de l'évolution du *framework*, il est probable que la prise en charge s'améliorera.

ESP32-C3 : une solution monocœur

L'ESP32-C3 est une alternative économique à l'ESP8266 et possède un grand nombre des périphériques de l'ESP32. L'adaptateur USB/série et JTAG intégré permet d'échanger facilement des fichiers et des données par un port USB. Vous pouvez même commencer à déboguer le code

LIENS

[1] @ElektorMathias sur Twitter : https://twitter.com/ElektorMathias/status/1386623477604626433
[2] Notes du labo d'Elektor, 06/2021 : https://www.elektormagazine.fr/news/compte-rendu-elektor-lab-juin-2021
[3] Tweet sur l'adaptateur USB-série / JTAG de l'ESP32-C3 : https://twitter.com/ElektorMathias/status/1392475706647584776
[4] Discussions du forum sur les révisions de l'ESP32-C3 : https://esp32.com/viewtopic.php?t=21040
[5] Page Wikipédia (en anglais) sur Cadence Tensilica : https://en.wikipedia.org/wiki/Tensilica
[6] Schémas de l'ESP32 C3 DevKitC-02 : https://dl.espressif.com/dl/schematics/SCH_ESP32-C3-DEVKITC-02_V1_1_20210126A.pdf
[7] Chargeur de fichiers dans ESP32 pour EDI Arduino (version corrigée) : https://github.com/lorol/arduino-esp32fs-plugin/releases
[8] Chargeur de fichiers dans ESP32 pour EDI Arduino : https://github.com/me-no-dev/arduino-esp32fs-plugin
[9] Dépôt Github – serveur mini-NTP avec GPS : https://github.com/ElektorLabs/180662-mini-NTP-ESP32
[10] Serveur mini-NTP avec GPS, Elektor Labs : https://www.elektormagazine.fr/labs/mini-ntp-server-with-gps
[11] Dépôt GitHub de cet article : https://bit.ly/3CD6iNs

(tant que rien d'imprévu ne se produit avec la révision 3 de la puce).

Grâce au *framework* Arduino, le code existant peut être réutilisé avec l'ESP32-C3 et le volume généreux de mémoire RAM et Flash permet d'envisager des projets plus conséquents. Concernant l'avenir, nous sommes sûrs de voir toute une série de cartes ESP32-C3 tierces commencer à apparaître dans les points de vente européens. Il sera intéressant de voir quelles nouvelles fonctions ces cartes offriront. Les possesseurs d'une carte ESP32-DevKitC-02 dans leur labo personnel seront bien placés pour commencer à écrire et tester immédiatement du code pour cet environnement. Les listages des croquis Arduino utilisés ici sont accessibles sur notre page GitHub [11].

210466-04

Listage 2. Lecture des informations relatives à la puce.

```
/* From the IDF documentation at https://github.com/espressif/esp-idf/components/esp_hw_support/include/
esp_chip_info.h */
 /*
  typedef enum {
     CHIP_ESP32  = 1, //!< ESP32
     CHIP_ESP32S2 = 2, //!< ESP32-S2
     CHIP_ESP32S3 = 4, //!< ESP32-S3
     CHIP_ESP32C3 = 5, //!< ESP32-C3
     CHIP_ESP32H2 = 6, //!< ESP32-H2
 } esp_chip_model_t;
// Chip feature flags, used in esp_chip_info_t
 #define CHIP_FEATURE_EMB_FLASH      BIT(0)      //!< Chip has embedded flash memory
 #define CHIP_FEATURE_WIFI_BGN       BIT(1)      //!< Chip has 2.4GHz WiFi
 #define CHIP_FEATURE_BLE            BIT(4)      //!< Chip has Bluetooth LE
 #define CHIP_FEATURE_BT             BIT(5)      //!< Chip has Bluetooth Classic
 #define CHIP_FEATURE_IEEE802154     BIT(6)      //!< Chip has IEEE 802.15.4
typedef struct {
     esp_chip_model_t model;  //!< chip model, one of esp_chip_model_t
     uint32_t features;       //!< bit mask of CHIP_FEATURE_x feature flags
     uint8_t cores;           //!< number of CPU cores
     uint8_t revision;        //!< chip revision number
 } esp_chip_info_t;
*/

void setup() {
     Serial.begin(115200);
 }

void loop() {
     delay(5000);
     Serial.println("esp_chip_info()");
     Serial.println("----------------------------------------------------------------");
     esp_chip_info_t info;
     esp_chip_info(&info);
     Serial.print("Chip Model: ");
     switch(info.model){
         case 1:{
             Serial.println("ESP32");
         }break;
        case 2:{
             Serial.println("ESP32-S2");
         }break;
```

Des questions, des commentaires ?

Envoyez un courriel à l'auteur (mathias.claussen@elektor.com) ou contactez Elektor (redaction@elektor.fr).

Contributeurs

Texte et illustrations : **Mathias Claußen**
Rédaction : **Jens Nickel, C. J. Abate**
Mise en page : **Giel Dols**
Traduction : **Pascal Godart**

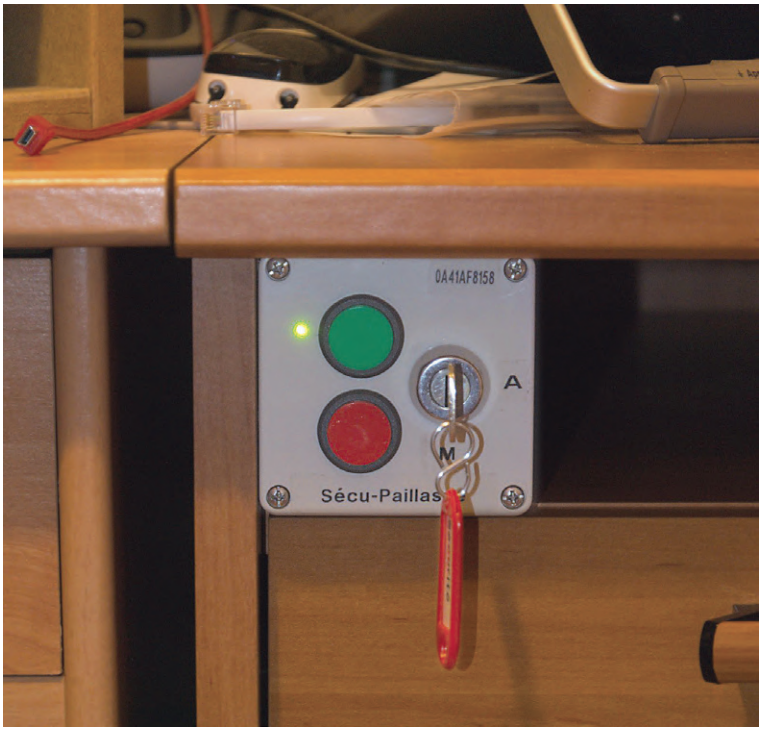
PRODUITS

- **Kit de développement ESP-C3-12F avec 4 Mo de mémoire Flash**
  www.elektor.fr/19855

```
        case 4:{
             Serial.println("ESP32-S3");
         }break;
        case 5:{
             Serial.println("ESP32-C3");
         }break;
        case 6:{
             Serial.println("ESP32-H2");
         } break;
        default:{
             Serial.print("Unknown Chipmodel");
             Serial.println(info.model);
         }
     }

    Serial.print("Featues :");
     if(info.features&CHIP_FEATURE_EMB_FLASH){
         Serial.print(" Embedded Flash ");
     }
     if(info.features&CHIP_FEATURE_WIFI_BGN){
         Serial.print(" WiFi(BGN) ");
     }
     if(info.features&CHIP_FEATURE_BLE){
         Serial.print(" BLE ");
     }

     if(info.features&CHIP_FEATURE_BT){
         Serial.print(" BT CLassic ");
     }
    if(info.features&CHIP_FEATURE_IEEE802154){
         Serial.print(" IEEE802.155.4 ");
     }
     Serial.println("");
     Serial.print("Cores: ");
     Serial.println(info.cores);
     Serial.print("Chip Revision: ");
     Serial.println(info.revision);
     Serial.println("-----------------------------------------------------------------");
     Serial.println();
}
```

protégez-vous et protégez les autres !

Interrupteur général pour paillasse (DIY)



Philippe Le Guen (France)

Sans surveillance, votre laboratoire d'électronique est-il sûr pour les visiteurs ? Vous, vous savez ce que vous faites, mais qu'en est-il de votre famille et de vos amis ? Risquent-ils l'électrocution en touchant quelque chose involontairement ? Écartez tout risque avec cet interrupteur général.

Pourquoi ai-je jugé nécessaire de rendre plus sûre ma paillasse ? Il y a diverses raisons, dont certaines d'ordre personnel. Tout d'abord, je veux pouvoir allumer et éteindre d'un seul geste tous les équipements de l'établi (instruments comme appareils en développement ou en réparation). En outre, en cas de coupure de courant accidentelle ou non, je ne veux pas que l'établi soit réalimenté automatiquement, je préfère le faire manuellement.
Enfin, après le départ de mon fils, j'ai installé mon petit labo dans son ancienne chambre, accessible à mes petits-enfants. Comme leurs petites mains explorent tout et touchent à tout, j'ai préféré protéger l'alimentation de l'établi

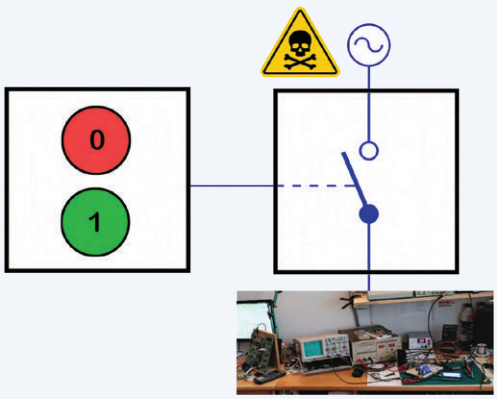


Figure 1. Le circuit est divisé en deux parties, le boîtier de commande (à gauche) et le boîtier à relais (à droite). Le boîtier de commande comporte aussi un interrupteur général de verrouillage à clé.

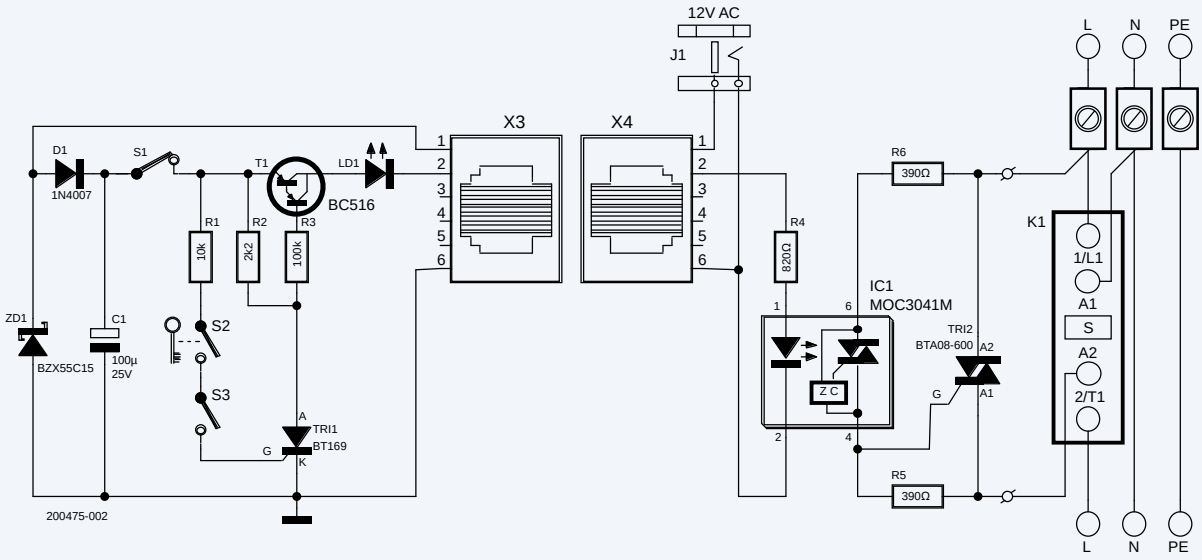


Figure 2. La partie gauche du circuit, le boîtier de commande, est alimentée par la partie droite, le boîtier à relais.

Un mot sur la coupure du secteur

Note de l'éditeur : dans tous les cas, couper le neutre (conducteur N) pour se déconnecter du réseau n'est pas judicieux ni autorisé, car l'énergie stockée dans un consommateur déconnecté (par ex. inductif) ne peut pas, dans ce cas, s'écouler via le conducteur N. C'est pourquoi seul un contacteur unipolaire est représenté dans le schéma de câblage. Si l'on veut quand même couper le conducteur N, il faut utiliser un contacteur avec un contact N en amont/aval.

par une clé. La mise hors tension doit être possible à tout moment.

Spécifications

Par sécurité, le circuit de commande doit être en basse tension (12 V) et isolé galvaniquement du commutateur par un optocoupleur. Le tout doit être facile à construire et à réparer donc un circuit électronique DIY à base de composants standard.

J'aurais pu utiliser un interrupteur de sécurité à déclenchement sans tension (*NVR*) du commerce (usité sur les machines-outils), mais je voulais le concevoir moi-même et qu'il puisse s'intégrer au tableau électrique que j'ai installé dans ce but sur mon établi.

Une conception en deux parties

J'ai conçu un système électronique simple n'utilisant que des composants standard (**fig. 1**). Un petit boîtier en polycarbonate abrite les boutons marche (vert) et arrêt (rouge) de l'établi ainsi que l'interrupteur à clé. J'avais déjà ce boîtier et il s'intégrait parfaitement sur l'établi, d'où mon choix.

J'ai récupéré l'interrupteur de verrouillage, je n'ai donc pas sa référence. Il doit être de type permanent, et verrouillable dans les deux positions. Voir cet exemple [1].

Un boîtier fixé sur l'un des rails DIN du tableau électrique abrite le relais secteur. Il est assisté d'un contacteur 230 V, 20 A (K1) pour le tarif « heures creuses ». On peut ainsi allumer l'établi manuellement en cas de défaillance du circuit de commande (qui sait ce qui peut arriver ?). J'en avais un qui traînait !

La **figure 2** montre le circuit complet. Le boîtier de commande est situé à distance du boîtier avec le contacteur et son interface, ils sont reliés par un câble équipé de fiches RJ11 aux deux extrémités. Le câble utilisé est de type « Ethernet cat. 5 » standard.

Alimentation basse tension

Le câble RJ11 achemine l'alimentation 12 V CA et le signal de sortie du boîtier de commande.

La diode D1 redresse la tension alternative et le condensateur C1 la filtre. La diode Zener de 15 V (ZD1) protège contre les surtensions. L'alimentation de 12 V CA provient d'un adaptateur de courant 12 V CC dont le redresseur a rendu l'âme. Seul le transformateur avec isolation de classe II est utilisé. On peut lui substituer une alimentation de 12 V CC. La diode D1 servirait alors de protection contre l'inversion de polarité. Le condensateur C1 assure le filtrage. Bien sûr, rien n'empêche d'augmenter sa valeur pour réduire la sensibilité du circuit au bruit et aux parasites.

Mémoire à thyristor

Après avoir appuyé sur le bouton vert S3, l'état *Marche* est mémorisé au moyen d'un thyristor (TRI1) qui pilote le transistor Darlington T1 et l'optocoupleur OK1. L'impulsion produite par le bouton-poussoir traverse la résistance R1 et TRI1 devient passant. Il le reste jusqu'à la disparition de son courant de charge à travers R2 ; c'est possible si on applique une

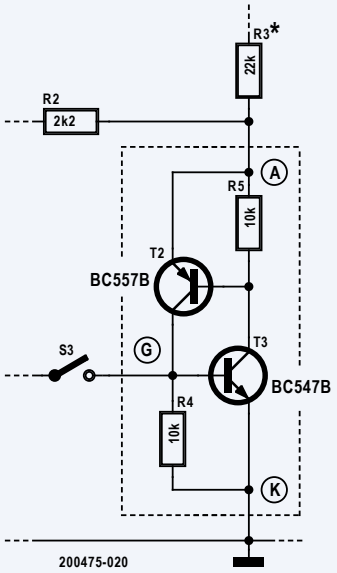


Figure 3. Une paire de transistors PNP/NPN peut remplacer le thyristor.

impulsion négative sur sa grille, ou si on coupe son alimentation. Le bouton-poussoir rouge *Arrêt* met à profit cette dernière possibilité.
Quand TRI1 est passant, le potentiel de la base de T1 baisse et T1 devient aussi passant. R3 polarise T1, tandis que R2 limite le courant traversant le thyristor. La LED LD1 et celle d'OK1 s'allument toutes deux, activant le triac TRI2 qui, à son tour, active le relais secteur. R4 limite le courant traversant les LED à 10 mA environ.
Quand on appuie sur le bouton rouge (S1), la tension d'alimentation de TRI1 disparaît. TRI1 se bloque immédiatement et entraîne le blocage de T1. Le courant dans la diode d'OK1 est coupé et le triac TRI2 cesse de conduire, désactivant ainsi le relais secteur. Notez que S1 doit être du type normalement fermé (*NC*).
L'optocoupleur OK1 ne commute qu'au moment du passage par zéro de la tension secteur, lorsque le courant est nul. Pour R5 et R6, la fiche technique d'OK1 recommande les valeurs qui ont été choisies ici.

Pas de circuit d'amortissement

Un triac avec charge inductive (moteur ou relais) nécessite normalement un circuit amortisseur (*snubber*), d'habitude constitué d'un réseau RC de faible valeur placé entre les deux anodes du triac. Ce n'est pas nécessaire ici car le TRI2 est un type dit *snubberless*.
Votre œil exercé aura remarqué l'absence de fusibles et de disjoncteurs. C'est que l'alimentation du boîtier à relais provient d'un interrupteur différentiel de 30 mA, et la sortie du relais secteur est dotée d'un disjoncteur. Mais peut-être ai-je eu tort de faire ainsi. Rien ne vous empêche d'ajouter des fusibles, bien sûr.

Version à pseudo-thyristor

Je propose également une seconde version de mon projet qui utilise une paire de transistors PNP/NPN (BC557B et BC547B) à la place du thyristor TRI1. Avec deux résistances de polarisation de 10 kΩ, ces transistors forment un pseudo-thyristor (**fig. 3**).
À la mise sous tension, les deux sont bloqués. T1 est également bloqué et aucun courant ne traverse l'optocoupleur OK1. Quand on appuie sur le bouton-poussoir *Marche*, T3 se met à conduire et la base de T2 approche

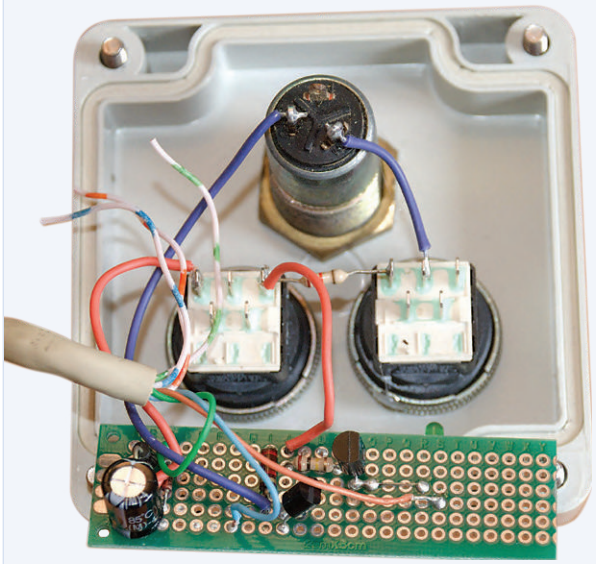


Figure 4. L'intérieur du boîtier de commande. Le petit circuit pastillé peut être fixé à côté des boutons-poussoirs avec une goutte de colle thermofusible.

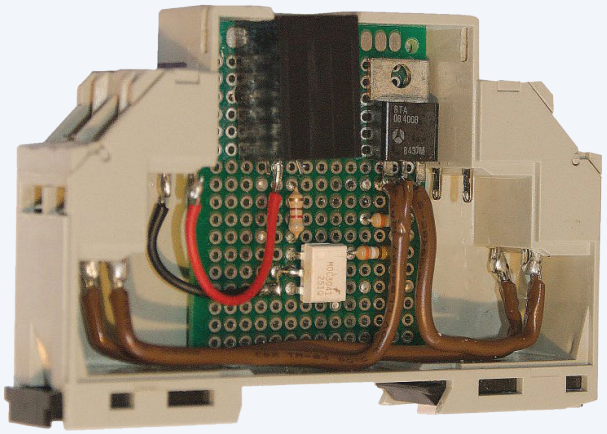


Figure 5. La partie à relais est intégrée dans un module de rail DIN. La pièce noire est le connecteur RJ11 auquel le boîtier de commande est relié.

de 0 V. T2 se met alors aussi à conduire et polarise positivement la base de T3, ce qui le maintient passant même après le relâchement du bouton-poussoir. R2 et R5 limitent le courant. Comme ci-dessus, un courant d'environ 10 mA commence à circuler dans OK1 via T1, également devenu passant. La mise hors tension est obtenue en appuyant sur le bouton-poussoir *Arrêt*.
Ce second circuit fonctionne exactement comme le premier, construit autour d'un vrai thyristor. Pour ma part, je préfère le thyristor, plus simple. En revanche, dans la seconde version, T1 n'a pas besoin d'être un Darlington et peut être remplacé par un BC557B dont il faut alors réduire la résistance de base à 22 kΩ. J'ai utilisé un BC516 car je ne trouvais pas mon sachet de BC557 dans mes tiroirs pourtant plutôt bien rangés.

Réalisation

Comme indiqué ci-dessus, j'ai utilisé deux boîtiers pour abriter le système en deux parties :

- Un boîtier presque carré en polycarbonate de 82×80×55 mm (Bopla M210, voir **fig. 4**). Ce boîtier abrite les deux boutons-poussoirs et l'interrupteur à clé ainsi que l'électronique de commande en 12 V. Ce boîtier s'adapte parfaitement à l'un de mes deux établis. Les boutons-poussoirs sont accessibles à tout moment depuis les deux établis.
- Un boîtier pour rail DIN de 17,5 mm de large pour accueillir l'étage secteur, dont l'optocoupleur et le triac (**fig. 5**). Il fournit également le 12 V AC du boîtier de commande sur une prise RJ11. Le triac n'a pas de dissipateur thermique en raison du faible courant.

Ce projet n'a pas eu besoin de circuit imprimé : un jeu de cartes perforées de prototypage acheté en ligne a fait l'affaire. Le circuit assemblé peut être fixé dans son boîtier avec un peu de colle thermofusible.

Gardez toujours à l'esprit que vous travaillez avec le secteur, donc soyez très prudent ! Les broches du triac sont assez proches les unes des autres !

Questions de sécurité

Le système présenté ici est simple et pratique à utiliser. Il rend votre paillasse beaucoup plus sûre, alors pourquoi ne pas en construire un pour votre labo privé ? La liste des composants peut être téléchargée ici [2].

200475-04

PRODUITS

- **Alimentation CA PeakTech 2240**
  www.elektor.fr/19315
- **Extracteur de fumée à éclairage LED**
  www.elektor.fr/19092

Contributeurs

Idée, conception, texte et illustrations : **Philippe Le Guen**
Schémas : **Patrick Wielders**
Rédaction : **Clemens Valens**
Mise en page : **Giel Dols**
Traduction : **Yves Georges**

Des questions, des commentaires ?

Contactez Elektor (redaction@elektor.fr).

LIENS

[1] Exemples d'interrupteurs à clé : https://www.mouser.com/datasheet/2/140/KO-345961.pdf
[2] Fichiers de conception, fiches techniques et liste des composants : https://www.elektormagazine.fr/labs/4209
[3] Site de l'auteur : https://www.pleguen.fr/index.php/securiser-sa-paillasse

2ème PARTIE

Création d'interfaces graphiques en Python

Mon nom n'est pas Bond

Créez une application graphique interactive avec la bibliothèque guizero.

MAKER

Laura Sach

Laura dirige l'équipe *A Level* de la Fondation RPi chargée des ressources pédagogiques en informatique à destination des étudiants.

@CodeBoom

MAKER

Martin O'Hanlon

Martin crée des cours, des projets et des ressources en ligne au sein de l'équipe *Learning* de la Fondation RPi.

@martinohanlon

Dans la première partie, nous avons appris à créer une fenêtre et à y positionner texte et image. Nous allons ici ajouter à cette fenêtre ce qui fait tout l'intérêt d'une interface graphique : sa capacité à interagir avec l'utilisateur. Pour cela nous créerons un gros bouton rouge générant des noms d'agents secrets.

Nous partirons de ce que vous savez déjà faire, à savoir créer une fenêtre principale (un widget App) contenant du texte (un widget Text). Voici notre code de départ (les lignes commençant par # sont des commentaires, spy = espion) :

```
# Imports ---------------
from guizero import App, Text

# Functions -------------

# App -------------------
app = App("TOP SECRET")

# Widgets ---------------
title = Text(app, "Push the red button to find out your spy name")

# Display ---------------
app.display()
```

Si vous l'exécutez, vous devriez voir la fenêtre reproduite sur la **figure 1**.

Ajout d'un bouton

Ajoutez aux classes App et Text importées en début de code la classe **PushButton** (notez les deux majuscules) qui est la classe permettant de créer des objets de type bouton. Pour en créer un, ajoutez la ligne suivante juste après la ligne instanciant le

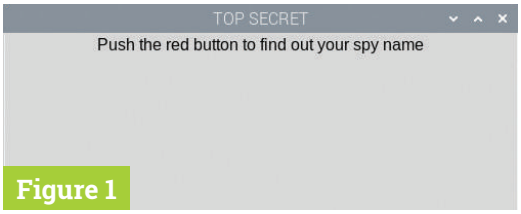
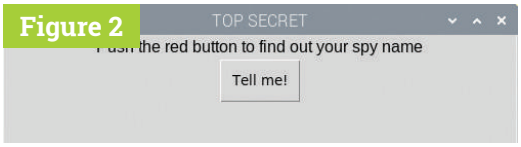


Figure 1

▶ **Figure 1** Affichage du widget **title** dans la fenêtre.

Figure 2



▲ **Figure 2** Création du bouton.

widget Text appelé **title** (listage **spy1.py**) :

```
button = PushButton(app, choose_name, text="Tell me!")
```

L'exécution de ce nouveau code ne crée pas de bouton, mais renvoie une erreur :

```
NameError: name 'choose_name' is not defined
```

De quoi se plaint l'interpréteur Python ? La plupart des éléments graphiques d'une interface peuvent être associés à une commande qui sera lancée lorsque l'utilisateur cliquera sur l'élément en question, un bouton p. ex. Ce que nous dit l'interpréteur est justement : vous m'avez donné un nom de commande (`choose_name`), mais je n'en trouve la définition nulle part ; revenez me voir quand vous aurez corrigé cette erreur. Cette commande est presque toujours le nom d'une fonction à exécuter, et c'est le cas ici : **choose-name** est le nom de la fonction qui générera des noms d'espion.

Définition de la fonction

Écrivons donc le code de la fonction `choose_name` qui sera lancée lorsque nous cliquerons sur le bouton.

Placer dans une même section toutes les fonctions associées aux widgets du programme améliore la lisibilité du code. C'est la raison pour laquelle notre squelette de programme contient une section *Functions*. Ajoutez-y ces deux lignes (listage **spy2.py**) :

```
def choose_name():
    print("Button was pressed")
```

Cette fois-ci le bouton apparaît (**fig. 2**). S'il ne se passe apparemment rien lorsque vous cliquez dessus, c'est que la commande `print()` de la

fonction affiche sa sortie dans la fenêtre de l'interpréteur de commandes, ou shell (**fig. 3**).

Utiliser `print()` dans la fonction représentant la commande associée au bouton est un moyen pratique et rapide de s'assurer que cette fonction est bien appelée. Cette vérification faite, on peut alors remplacer `print()` par le véritable code à déclencher.

Placez le caractère `#` au début de la ligne **print("Button was pressed")**. Lorsque l'interpréteur Python verra ce # au début d'une ligne, il l'ignorera entièrement et passera à la ligne suivante. C'est ce qu'on appelle « commenter » une ligne de code. Pourquoi laisser une ligne qui sera ignorée aussi sûrement que si elle n'existait pas ? Tout simplement pour éviter d'avoir à la retaper si elle devait servir à nouveau, puisque dans ce cas il suffira de supprimer le `#`.

spy1.py

TÉLÉCHARGEZ LE CODE COMPLET : magpi.cc/guizerocode

Langage : **Python 3**

```
001.  # Imports ---------------
002.  from guizero import App, Text, PushButton
003.
004.  # Functions -------------
005.
006.  # App -------------------
007.  app = App("TOP SECRET")
008.
009.  # Widgets ---------------
010.  title = Text(app, "Push the red button to find out your spy name")
011.  button = PushButton(app, choose_name, text="Tell me!")
012.
013.  # Display ---------------
014.  app.display()
```

spy2.py

Langage : **Python 3**

```
001.  # Imports ---------------
002.  from guizero import App, Text, PushButton
003.
004.  # Functions -------------
005.
006.  # App -------------------
007.  app = App("TOP SECRET")
008.
009.  # Widgets ---------------
010.  title = Text(app, "Push the red button to find out your spy name")
011.  button = PushButton(app, choose_name, text="Tell me!")
012.
013.  # Display ---------------
014.  app.display()
```

Un gros bouton rouge, disiez-vous ?

Pour l'instant le bouton n'est ni gros ni rouge ! Pour qu'il le devienne il faut utiliser les propriétés définissant la couleur de fond et la taille du texte d'un widget, ici le widget **PushButton**. Référez-vous pour cela au tutoriel précédent ou au code final (lignes 24 et 25).

Notez que certaines versions du système d'exploitation macOS interdisent de modifier la couleur d'un bouton. Vous devriez cependant être à même de modifier la taille du texte.

Figure 3

▲ **Figure 3** Sortie de `print()` dans le shell.

Un vrai nid d'espions

Créez une liste de prénoms sur une nouvelle ligne. Ajoutez autant de prénoms que souhaité en veillant à ce que chacun soit entouré de guillemets droits et séparé du suivant par une virgule. Une suite de caractères composée de lettres, chiffres et/ou signes de ponctuation entourés de guillemets droits est appelée une chaîne. Autrement dit, chaque prénom doit être une chaîne.

```
first_names = ["Barbara", "Woody",
"Tiberius", "Smokey", "Jennifer", "Ruby"]
```

Ajoutez de même une liste de noms de famille :

```
last_names = ["Spindleshanks", "Mysterioso",
"Dungeon", "Catseye", "Darkmeyer",
"Flamingobreath"]
```

Il nous faut maintenant une fonction capable d'extraire de façon aléatoire un prénom et un nom à partir de nos deux listes. Ce genre de fonction est si utile qu'elle fait partie du langage Python, plus précisément du module **random** (aléatoire). Son nom est **choice**, et nous l'importons donc en début de code avec :

```
from random import choice
```

La syntaxe de la fonction `choice()` est simple : on lui fournit le nom d'une liste, et elle renvoie de façon aléatoire un élément de cette liste. Nous pouvons dès lors joindre (les programmeurs disent « concaténer ») le prénom et le nom retournés par `choice()` à l'aide du symbole plus (`+`), sans oublier de les séparer par une espace :

```
spy_name = choice(first_names) + " " +
choice(last_names)
print(spy_name)
```

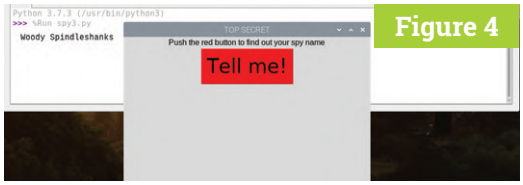


Figure 4

▲ **Figure 4** Affichage du nom de l'espion.

Enregistrez votre code (**spy3.py**) et lancez-le. Appuyez sur le bouton : vous devriez voir s'afficher un nom d'espion dans la fenêtre de l'interpréteur de commandes, là où s'était affichée la sortie de la commande **print("Button was pressed")** (**fig. 4**).

Un espion à la fenêtre

Il serait bien sûr plus agréable de voir le nom de l'espion s'afficher dans la fenêtre principale. Pour cela nous avons besoin d'un autre widget **Text**. Appelons-le **name** et ajoutons-le dans la section *Widgets* :

```
name = Text(app, text="")
```

Ce widget ne doit rien afficher pour l'instant puisque le nom de l'espion n'apparaîtra qu'après un clic sur le bouton. C'est pour cela que nous avons passé au paramètre **text** une chaîne vide (`text=""`).

Commentez la ligne **print(spy_name)** de la fonction `choose_name()` pour qu'elle ne s'affiche plus, puis ajoutez la ligne suivante à la fin de la fonction :

```
name.value = spy_name
```

Lorsque la fonction `choose_name` sera appelée, cette instruction affectera le contenu de la variable `spy_name` (le prénom et le nom de notre espion) à la valeur du widget **name**, c.-à-d. à son paramètre **text**.

Lancez le code final (**03-spy-name-chooser.py**) et cliquez sur le bouton pour voir un nom d'espion aléatoire s'afficher dans la fenêtre (**fig. 5**). Cliquez à nouveau sur le bouton si vous refusez de partir en mission affublé d'un nom pareil. **(VF : Hervé Moreau)**

Figure 5

▲ **Figure 5** La fenêtre créée par le code final.

spy3.py

Langage : **Python 3**

```
# Imports ---------------
from guizero import App, Text, PushButton
from random import choice

# Functions -------------
def choose_name():
    #print("Button was pressed")
    first_names = ["Barbara", "Woody", "Tiberius", "Smokey", "Jennifer", "Ruby"]
    last_names = ["Spindleshanks", "Mysterioso", "Dungeon", "Catseye", "Darkmeyer", "Flamingobreath"]
    spy_name = choice(first_names) + " " + choice(last_names)
    print(spy_name)

# App -------------------
app = App("TOP SECRET")

# Widgets ---------------
title = Text(app, "Push the red button to find out your spy name")
button = PushButton(app, choose_name, text="Tell me!")
button.bg = "red"
button.text_size = 30

# Display ---------------
app.display()
```

03-spy-name-chooser.py

Langage : **Python 3**

```
# Imports ---------------

from guizero import App, Text, PushButton
from random import choice

# Functions -------------

def choose_name():
    #print("Button was pressed")
    first_names = ["Barbara", "Woody", "Tiberius", "Smokey", "Jennifer", "Ruby"]
    last_names = ["Spindleshanks", "Mysterioso", "Dungeon", "Catseye", "Darkmeyer", "Flamingobreath"]
    spy_name = choice(first_names) + " " + choice(last_names)
    #print(spy_name)
    name.value = spy_name

# App -------------------

app = App("TOP SECRET")

# Widgets ---------------

title = Text(app, "Push the red button to find out your spy name")
button = PushButton(app, choose_name, text="Tell me!")
button.bg = "red"
button.text_size = 30
name = Text(app, text="")

# Display ---------------

app.display()
```

Note de l'éditeur : cet article est paru initialement dans le magazine MagPi (le magazine officiel du Raspberry Pi).
La maison d'édition Elektor publie ce magazine en néerlandais, allemand et français (www.magpi.fr).

productronica
fast forward
the start-up platform
powered by Elektor

gagnants du concours Fast Forward 2021 du salon productronica

Des technologies passionnantes et des solutions d'ingénierie créatives

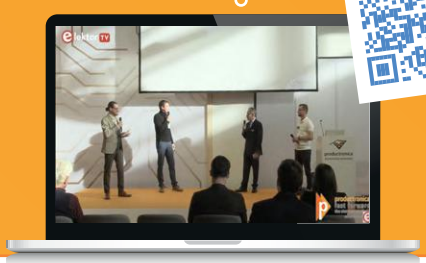
Clemens Valens (Elektor)

productronica fast forward 2021, la plateforme pour les jeunes pousses soutenue par Elektor, a offert aux entreprises novatrices une occasion unique de présenter leurs technologies à la communauté électronique mondiale. Voici les gagnants !

Il y a eu longtemps une incertitude sur les conditions de déroulement du salon productronica 2021. Serions-nous tous devant nos écrans, ou bien présents sur notre stand ? La seule façon d'avancer était donc de travailler en ligne. Par conséquent, au lieu de se présenter devant un jury et un public en direct, les candidats au Fast Forward Award du salon productronica 2021, originaires de pays tels que la Hongrie, l'Allemagne, Israël et la France, ont dû enregistrer leurs présentations et les soumettre en ligne. Chaque jeune pousse a également fourni des informations supplémentaires sous la forme de plans d'affaires et de dossiers de présentation.

Après avoir regardé les « elevator pitches » préenregistrés et étudié toute la documentation, le jury du Fast Forward Award du salon productronica 2021 – qui comprenait Benjamin Klingenberg (NCAB Group, sponsor « platine » de l'événement) et des représentants du labo et de la rédaction d'Elektor – a sélectionné les trois jeunes pousses gagnantes.

Regardez la cérémonie de remise des prix !



PREMIER PRIX

50.000 € de budget marketing chez Elektor

ioTech Group – Basée à Jérusalem (Israël), l'entreprise ioTech développe de nouvelles technologies pour la fabrication additive de masse et l'impression multi-matériaux.

www.elektormagazine.com/news/iotech-group-productronica-fast-forward-2021

DEUXIÈME PRIX

25.000 € de budget marketing chez Elektor

Reedu GmbH & Co. KG – L'entreprise Reengineering Education alias Reedu ou re:edu de Münster (Allemagne) est connue pour la gamme senseBox (station de mesures environnementales) et la plateforme openSenseMap.

www.elektormagazine.com/news/reedu-gmbh-co-kg-productronica-fast-forward-2021

TROISIÈME PRIX

15.000 € de budget marketing chez Elektor

Pozi Development Ltd – L'entreprise Pozi de Budapest (Hongrie) propose un système de surveillance de la fabrication et de la logistique, et identifie et élimine les gaspillages tout au long de la chaîne d'approvisionnement.

www.elektormagazine.com/news/pozi-development-ltd-productronica-fast-forward-2021

Les trois jeunes pousses gagnantes ont été choisies par le jury en raison du fort potentiel commercial de leur innovation et de la qualité de leurs équipes, deux critères importants pour qu'une jeune pousse devienne une entreprise prospère. Félicitations aux gagnants !

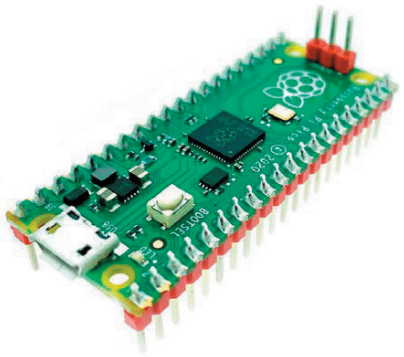
On se voit au salon electronica 2022 ?

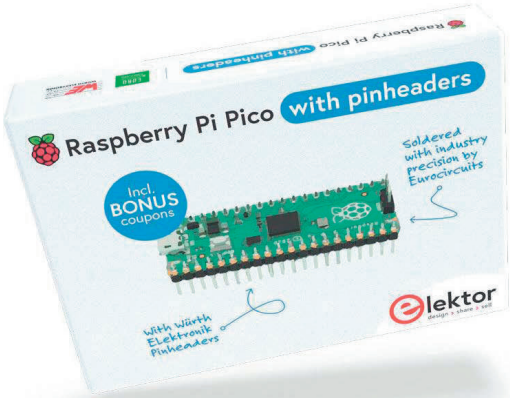
Le Fast Forward Award devrait être remis en jeu à l'occasion du salon electronica 2022 (https://electronica.de/de/), à Munich, en novembre de cette année. Espérons que tout sera rentré dans l'ordre d'ici là et que nous vous y verrons en personne !

210658-04

ABONNEZ-VOUS ET RECEVEZ

Raspberry Pi + Headers GRATUIT





Souscrivez dès maintenant un abonnement d'un an au magazine MagPi, nous vous offrons :

- Six numéros du magazine MagPi
- Une carte Raspberry avec headers

TOUS LES 2 MOIS, LES DERNIÈRES NOUVELLES DU RASPBERRY PI ET LES MEILLEURS PROJETS !

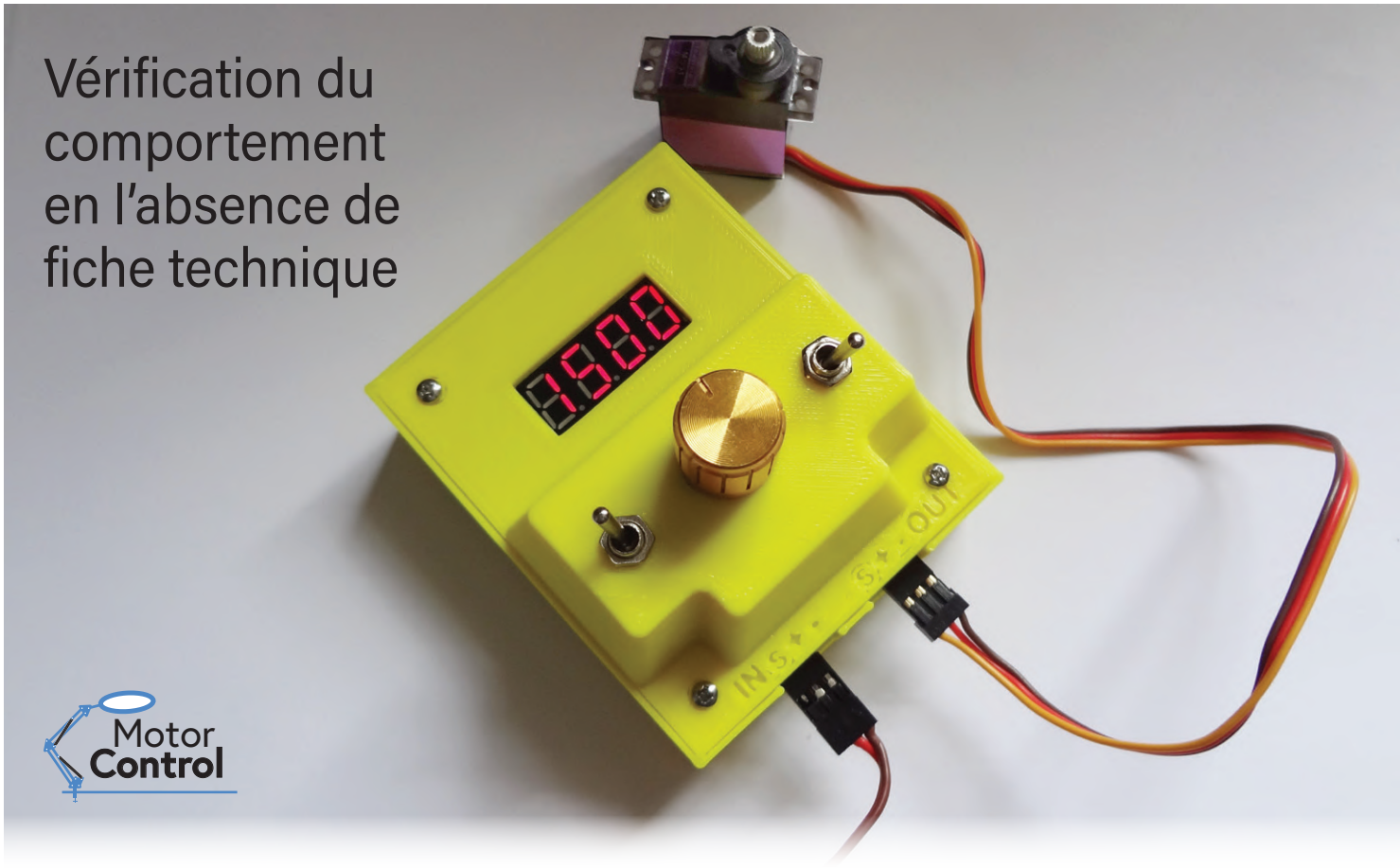
SEULEMENT 54,95 € PAR AN (6 NUMÉROS)

Vos avantages :

- Prix au numéro réduit
- Chaque numéro directement dans votre boîte aux lettres
- Tous les numéros disponibles sous forme numérique (PDF)
- Découverte de chaque nouveau numéro avant sa sortie en kiosque

ABONNEZ-VOUS : WWW.MAGPI.FR

testeur de servo polyvalent



Marcelo Maggi (États-Unis)

La connaissance précise du comportement d'un servo, surtout en cas d'absence de fiche technique, est incontournable pour garantir le fonctionnement de l'application finale, d'autant plus que prolifèrent de nos jours toutes sortes de servos de qualité, performance et prix divers.

On appelle servomécanisme, ou servo, un dispositif asservi dans une boucle de rétroaction, dont la grandeur de sortie est de nature mécanique. En électronique, et en particulier dans le domaine des actionneurs électromécaniques, un servomoteur est un organe qui traduit un signal électrique en un mouvement mécanique.

On rencontre ces servos dans de nombreux domaines. Ici, je m'intéresserai à ceux des maquettes radiocommandées, extrêmement populaires et largement utilisés, en modélisme comme dans d'autres activités de loisirs. Il existe de nombreux types de servos, analogiques et numériques, avec des engrenages en plastique ou en métal, avec ou sans roulements à billes, etc. Mais ils ont tous un point commun : ils peuvent être commandés par un même signal électrique.

Caractéristiques et spécifications

Modes : manuel (basse et haute résolution) et automatique (basse et haute vitesse).
Largeur d'impulsion minimale : réglable par logiciel.
Largeur d'impulsion maximale : réglable par logiciel.
Position centrale : peut être réglée manuellement.
Résolution minimale du pas/impulsion : 1 µs.
Pas maximum : réglable par logiciel.
Période du signal : fixée à 20 ms.
Stabilité/précision de l'impulsion : contrôlée par quartz.
Alimentation électrique : 4,5 à 6,0 V.

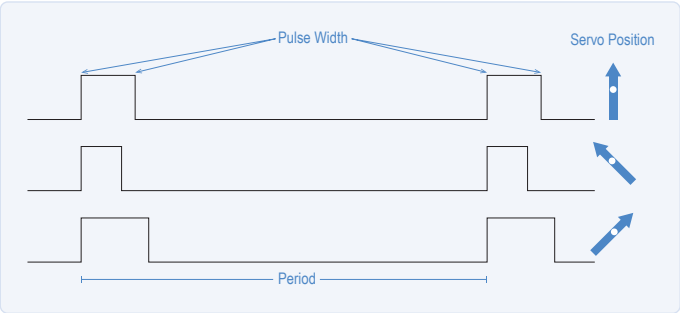


Figure 1. Impulsions de commande du servo.

Ce signal est constitué d'impulsions de largeur variable, séparées par une période fixe (**fig. 1**), généralement de vingt millisecondes (ms). La largeur des impulsions est ce qui contrôle effectivement la rotation du servo. La position centrale est fixée à 1500 microsecondes (µs) pour la plupart des servos, tandis qu'on atteint les positions extrêmes à 500 µs minimum pour l'une et 2500 µs maximum pour l'autre. Habituellement, ces extrêmes correspondent à une rotation de −90° et +90° à partir du point central, le sens de rotation dépendant du fabricant.
L'amplitude de l'impulsion, en volts, est généralement égale à la tension d'alimentation du servo dont les plus courantes sont 4,8 V et 6,0 V, mais on en trouve aussi des amplitudes de 3,7 V à 14 V et plus.

Pourquoi un testeur de servo ?

La fiche technique du fabricant censée donner toutes les caractéristiques du servo (angle maximum de rotation, vitesse, bande morte, etc.) n'existe pas toujours, surtout pour la plupart des nombreux servos à bas prix proposés sur le marché. Même quand certaines données sont disponibles, il est fréquent que la mesure des caractéristiques réelles donne des résultats très différents, et ce qui compte en définitive, ce sont les performances réelles de votre servo, et non son comportement théorique.

Conception d'un testeur de servo

Il est très facile de construire un testeur de servo simple, et il y en a beaucoup d'exemples sur le web. Il suffit d'un 555 (un circuit intégré très populaire) et de quelques autres composants pour réaliser un testeur de servo analogique. L'étape suivante serait un appareil avec un contrôle précis de la largeur d'impulsion et un affichage de cette valeur. On peut en trouver des versions bon marché sur le web, mais elles présentent un inconvénient : la plupart des paramètres sont déjà programmés et ne peuvent pas être modifiés par l'utilisateur. Ce serait beaucoup mieux de disposer d'un testeur de servo entièrement personnalisable, dans lequel tous les paramètres seraient accessibles à l'utilisateur pour créer le signal souhaité, y compris les fonctions de test manuel et automatique.
C'est ce qui est proposé ici : un testeur entièrement programmable donnant accès à tous les paramètres du signal, avec les modes manuel et automatique et une interface utilisateur claire et intuitive.

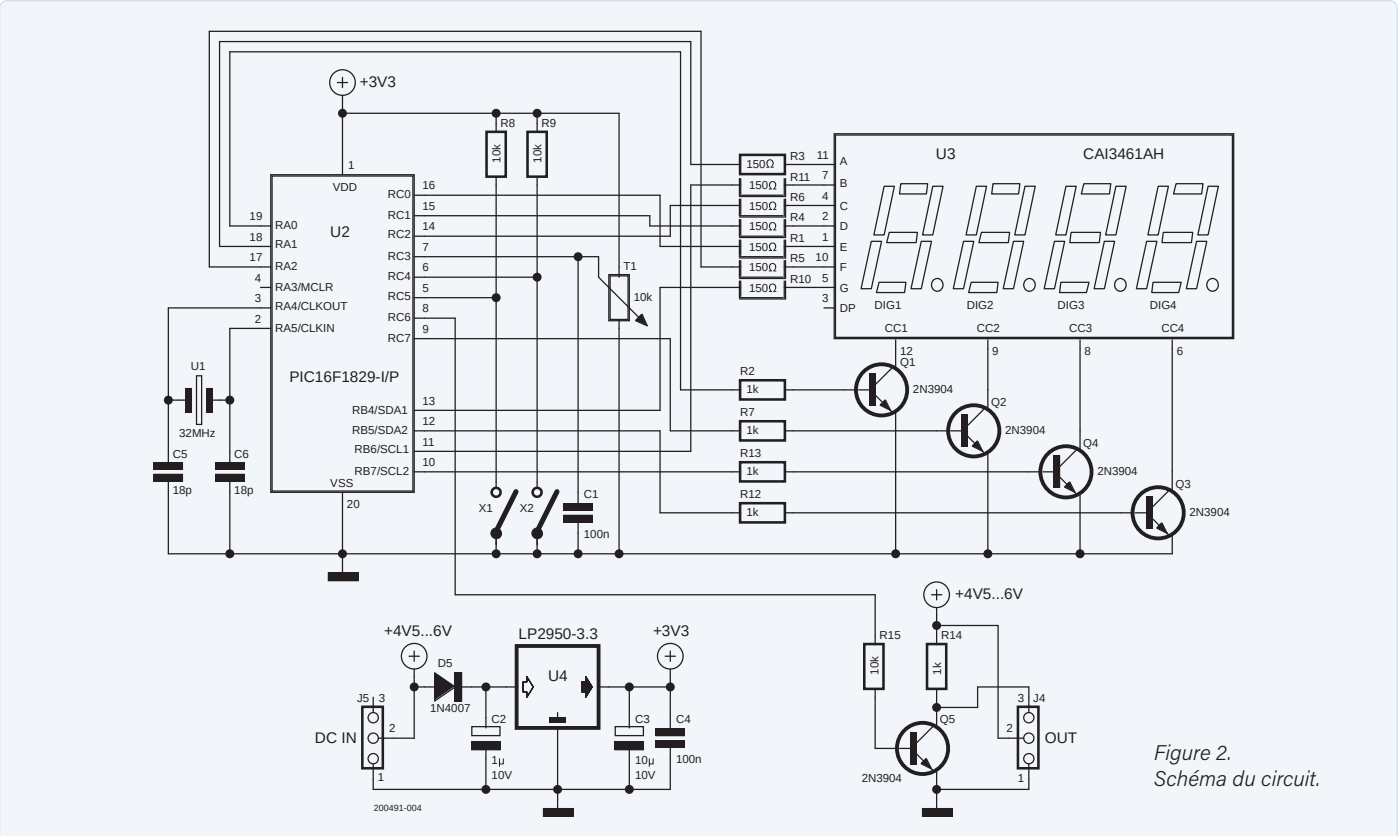


Figure 2. Schéma du circuit.

Matériel informatique

Le schéma complet du circuit est présenté à la **figure 2**. Les connexions externes sont indiquées dans la partie inférieure ; à gauche se trouve l'entrée d'alimentation (J5 - DC IN), qui peut être connectée à l'alimentation de la maquette radiocommandée, ou à toute autre alimentation capable de fournir le courant requis pour le servo testé. L'unité consomme moins de 15 mA, un supplément de charge faible pour l'alimentation. La tension peut être comprise entre 4,5 et 6,0 V, selon les besoins du servo.

Remarque : le servo recevra la pleine tension de la source d'alimentation. Veillez à ne pas dépasser les valeurs maximales du servo ! Pour se conformer aux spécifications de la plupart des fabricants, le connecteur central est toujours la borne +V, tandis que les extrêmes sont GND et SIGNAL.

L'entrée d'alimentation ne nécessite que deux lignes, +V et GND, mais un connecteur à 3 bornes a été monté pour recevoir n'importe quel connecteur ESC standard. Sur le côté droit du schéma se trouve la sortie du servo (J4 - OUT), avec les trois lignes connectées (+V, GND et SIGNAL), à connecter au servo en cours de test.

La partie supérieure-milieu gauche montre le microcontrôleur (U2) et l'interface utilisateur, un seul potentiomètre (T1) et deux interrupteurs simples (X1 et X2). La rotation du potentiomètre modifie la durée de l'impulsion tandis que les commutateurs sélectionnent les quatre modes de fonctionnement, comme le montre ce tableau (les différents modes seront expliqués dans la section consacrée au logiciel) :

SW1	SW2	Mode
0	0	Auto, haute vitesse
0	1	Auto, basse vitesse
1	0	Manuel, haute résolution
1	1	Manuel, faible résolution

Enfin, en haut à droite se trouve U3 : l'afficheur à 4 chiffres à 7 segments et ses composants.

Dessin de la carte

Bien que ce matériel puisse fonctionner s'il est monté sur une platine d'essai, un circuit imprimé le rendra plus durable et plus fiable. Un tracé possible de circuit imprimé est présenté sur la **figure 3**. Les fichiers de conception (NI Multisim) et les fichiers Gerber pour commander la carte chez votre fournisseur préféré sont disponibles au téléchargement sur [1]. Notez que les condensateurs du circuit oscillant du microcontrôleur sont absents de cette version du circuit imprimé ; apparemment le testeur de servo de l'auteur se débrouille très bien sans C5 et C6. Les condensateurs peuvent facilement être ajoutés sur le côté cuivre (inférieur) de la carte, vous devrez peut-être gratter le vernis épargne du plan de masse. Mais bien sûr, vous pouvez aussi télécharger les fichiers de conception et ajouter les deux condensateurs au tracé.

Logiciel

La partie la plus importante de ce montage est sans doute le logiciel, qui lui apporte la précision et la souplesse. Le programme s'exécute dans un PIC16F1829, cadencé à 32 MHz ; il est écrit en C, traduit par le compilateur C de CCS. Le code source et le fichier HEX pour la programmation du microcontrôleur peuvent être téléchargés sur [1].

Le programme commence par les définitions du matériel et des connexions, ainsi que des valeurs de référence clés : les lettres `a` à `g` représentent les 7 segments de l'affichage numérique, tandis que `d1` à `d4` correspondent aux quatre chiffres. `PWM` est le signal de sortie pour piloter le servo et `SW1` / `SW2` sont les interrupteurs. Le potentiomètre est connecté à l'entrée analogique AN7 (PIN_C3) et il est réglé dans la fonction `main`.

LISTE DES COMPOSANTS

Résistances

R1,R3,R4,R5,R6,R10,R11 = 150 Ω
R2,R7,R12,R13,R14 = 1 kΩ
R8,R9,R15 = 10 kΩ
T1 = potentiomètre 10 kΩ

Condensateurs

C1,C4 = 100 nF
C2 = 1 µF/10 V radial
C3 = 10 µF/10 V radial
C5,C6 = 18 pF

Semi-conducteurs

D5 = 1N4007
Q1,Q2,Q3,Q4,Q5 = 2N3904
U2 = PIC16F1829-I/P
U3 = afficheur à LED, CC, à 4 chiffres et 7 segments, 3461AH
U4 = régulateur de tension LDO 3,3 V LP2950-3.3

Divers

U1 = quartz de 32 MHz
X1,X2 = interrupteur simple unipolaire

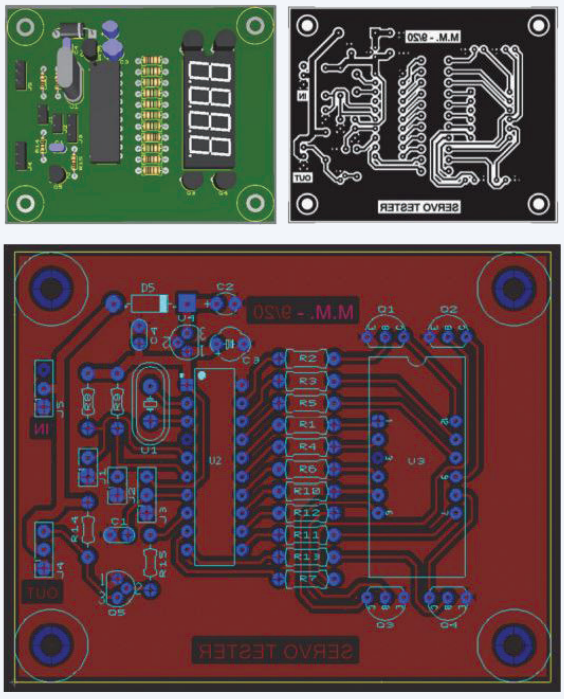


Figure 3. Dessin de la carte équipée ; le potentiomètre est J3, et les interrupteurs sont J1 et J2.

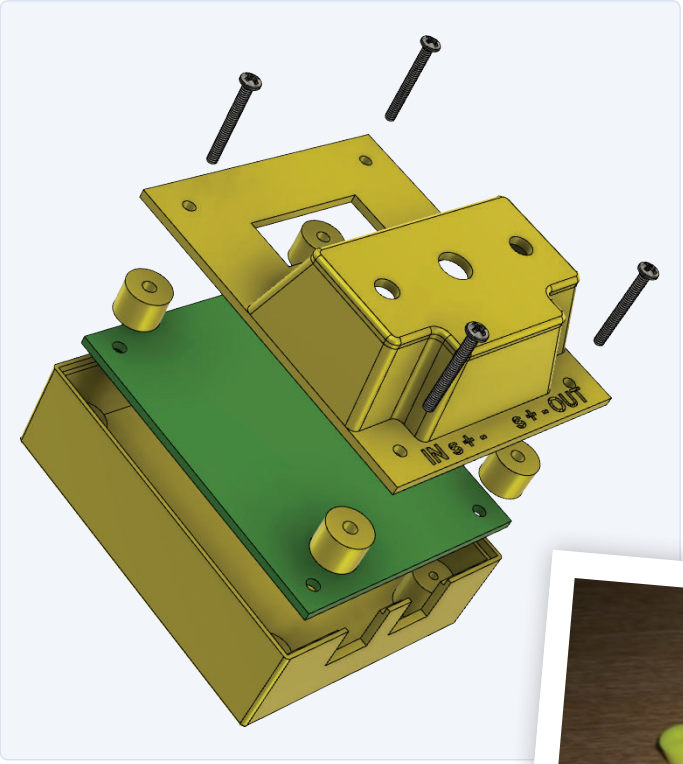


Figure 4. Conception 3D du boîtier.

Ces quatre paramètres confèrent au testeur une grande polyvalence, qu'on ne retrouve sur aucun appareil commercial bon marché. Si vous montez le microcontrôleur sur une embase, vous pouvez facilement le retirer, le reprogrammer et disposer d'un nouveau testeur conforme à vos besoins ; ou si vous êtes satisfait d'un jeu de paramètres, ne touchez plus à rien.

Chaque chiffre est entièrement défini à l'intérieur d'une fonction (`num`) et peut être sélectionné à l'aide d'une instruction `switch` ; chaque fois que cette fonction est appelée avec un nombre stocké dans `n`, le nombre sera allumé sur l'afficheur. Pour sélectionner lequel des quatre chiffres est actif à un moment donné (puisque l'affichage est balayé séquentiellement), la fonction `digit` inclut la sélection, également dans une instruction `switch`. Chaque fois qu'il est appelé, le chiffre stocké dans `z` s'allume. Si `z` contient « 5 », alors tous les chiffres seront éteints ; ceci est utilisé pour avoir une transition propre d'un chiffre à l'autre, sans rémanence dans l'affichage.

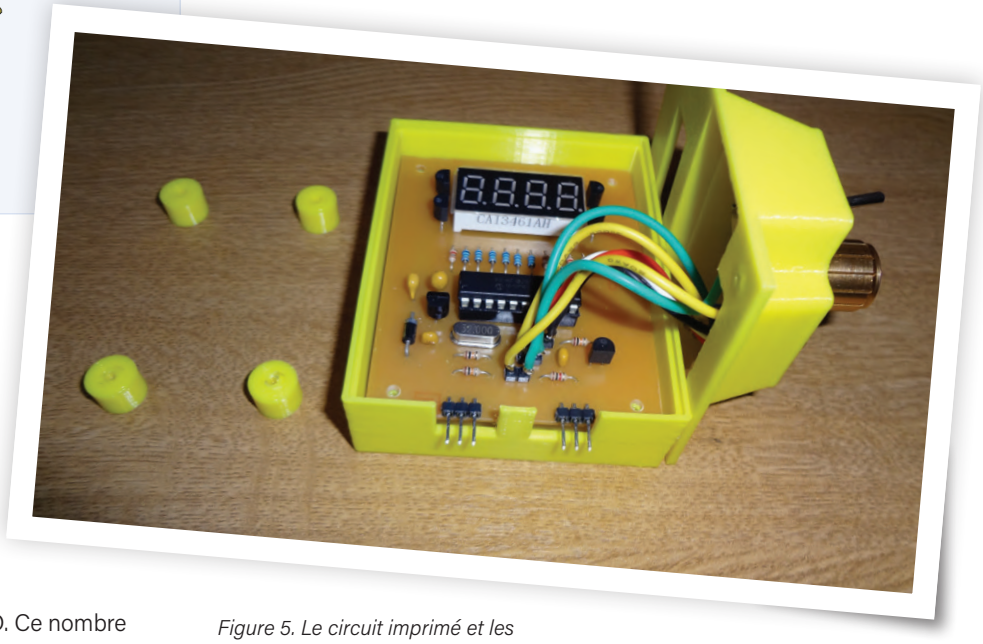


Figure 5. Le circuit imprimé et les connexions à l'intérieur du boîtier.

En bas, il y a quatre définitions très importantes :

- BRI : luminosité (*brightness*) de l'afficheur à LED. Ce nombre commande la persistance de chaque chiffre (en µs), donc sa luminosité apparente. Il est fixé à 50 mais peut être modifié de 1 à 200.
- SPD : vitesse (*speed*) du mouvement du servo en mode automatique. Ce nombre représente la taille du pas (en µs) lors de l'incrémentation/décrémentation de la largeur d'impulsion ; plus le nombre est élevé, plus le servo se déplace rapidement. Tout nombre compris entre 1 et 100 est valide.
- MIN : largeur d'impulsion minimale (en µs). Valeur la plus courte de l'impulsion en mode manuel et en mode automatique. Bien que n'importe quel nombre soit valide, il est recommandé de ne pas descendre en dessous de 500, car le servo pourrait ne pas suivre en deçà.
- MAX : largeur d'impulsion maximale (en µs). Valeur la plus longue de l'impulsion en mode manuel et en mode automatique. Bien que n'importe quel nombre puisse convenir, il est recommandé de ne pas dépasser 2500, car le servo pourrait ne pas suivre au-delà.

La fonction qui réunit tous les éléments (sélection des chiffres et sortie des nombres) est `display` ; cette fonction prend un nombre à 4 chiffres, le décompose en chiffres individuels et les allume en séquence sur l'afficheur, de droite à gauche (unités en premier), en appelant les deux fonctions précédentes. C'est ici qu'est contrôlée la luminosité de l'affichage, en réglant le délai avant d'effacer un chiffre.

Le chronométrage précis de la période et de la durée de l'impulsion est obtenu en utilisant l'interruption de TIMER1. Ce compteur, d'une largeur de 16 bits, déclenche une interruption quand il déborde à 65 536 ; en le chargeant avec une valeur initiale, un temps précis peut être mesuré

en traitant cette interruption. En séquence, la première fois que cette INT est appelée, TIMER1 est chargé avec la valeur correspondant à la durée de l'impulsion ; la fois suivante, ce sera le reste de la période totale. Ce processus est répété en continu (contrôlé par la variable `pulse`), de sorte qu'on obtient une largeur d'impulsion précise, tout en gardant une période constante de 20 ms.

Ici, nous vérifions également l'état des commutateurs pour décider si nous sommes en mode *automatique* (SW1 = 0). Si c'est le cas, SW2 va déterminer la vitesse du balayage automatique (de MIN à MAX et retour à MIN en continu) : SW2 = 1 signifie une vitesse normale (l'impulsion varie de 1 µs à la fois) et SW2 = 0 signifie une vitesse élevée (l'impulsion varie de SPD µs à la fois). Souvenez-vous que SPD peut être un nombre compris entre 1 et 100 ; s'il est réglé sur 1, la vitesse normale et la vitesse élevée sont identiques (ce qui n'est pas très utile).

Enfin, `main` rassemble tout ; outre l'initialisation du convertisseur analogique-numérique (CA/N), de TIMER1 et des interruptions, il contient la boucle du programme principal. Cette boucle lit le CA/N (donc la valeur du potentiomètre) et en déduit la durée de l'impulsion.

Si nous sommes en mode manuel (SW1 = 1), alors SW2 est évalué ; si « 1 », le mode *basse résolution* est actif, et le potentiomètre est capable de balayer toute la gamme de largeurs d'impulsion (MIN - MAX). Ce mode est appelé « basse résolution » pour une raison simple : le CA/N ne peut acquérir que 1 024 niveaux différents, mais la gamme de largeurs d'impulsion peut être de 1400 µs ou plus, il n'est donc pas possible d'obtenir une résolution de 1 µs dans la gamme complète. Si SW2 = 0, alors le mode *haute résolution* est activé. Ici, la gamme complète de largeurs d'impulsion est limitée de 1469 à 1531 µs. Le potentiomètre peut facilement être déplacé pour changer 1 µs à la fois, ce qui est extrêmement utile pour :

- Positionner un servo au point central (1500 µs).
- Mesurer la bande morte : de combien de µs l'impulsion doit-elle changer pour déplacer le servo ?

Donner un aspect professionnel

Un boîtier adéquat donnera à ce montage un aspect très professionnel et le protégera. Un simple boîtier imprimé en 3D (**fig. 4**) fera l'affaire, dont le fichier de conception peut être téléchargé sur [1]. Le circuit imprimé est « pris en sandwich » entre quatre entretoises et la coquille inférieure ; la coquille supérieure est fixée à l'aide de quatre vis Phillips Pan Head M2,5 × 20 mm (#2-56 × ¾). L'assemblage final est illustré à la **figure 5**.

200491-04

Contributeurs

Texte : **Marcelo Maggi**
Illustrations : **Marcelo Maggi, Patrick Wielders**
Rédaction : **Luc Lemmens**
Traduction : **Helmut Müller**
Mise en page : **Giel Dols**

Des questions, des commentaires ?

Contactez Elektor (redaction@elektor.fr).

PRODUITS

- **Mini servo à rotation continue de 360° - FS90R de FeeTech**
  www.elektor.fr/19784

- **Servo à engrenages en métal avec accessoires - FS90MG de FeeTech**
  www.elektor.fr/19788

LIEN

[1] Téléchargements de logiciels, de circuits imprimés et de conceptions 3D : http://www.elektormagazine.fr/200491-04

Modbus sans fil (partie 2)

Logiciel de la carte Modbus TCP sans fil

Josef Bernhardt et **Martin Mohr** (Allemagne)

Construit autour d'un module *NodeMCU d'Espressif*, la carte décrite dans la première partie de cette série permet d'utiliser le protocole *Modbus* sur un réseau sans fil. Cet article couvre le logiciel et la configuration de la carte Modbus.

Dans la 1ère partie [1] de cette série d'articles, nous avons abordé le matériel du projet. La carte est bâtie autour d'un module NodeMCU équipé d'un µcontrôleur *ESP8266*, avec une carte additionnelle fournissant des ports de type industriel. Après une brève introduction au principe du protocole Modbus, nous étudions ici la façon dont la carte est pilotée et le logiciel utilisé à cet effet. La commande de porte d'ascenseur de la 1ère partie sert à nouveau d'exemple.

Le protocole Modbus

Le protocole Modbus est largement utilisé dans l'automobile. Il fonctionne selon le principe maître/esclave. Le maître du bus peut commander jusqu'à 246 esclaves. Les adresses 1 à 247 peuvent être attribuées aux nœuds. L'adresse 0 est réservée aux données de diffusion ; tous les nœuds reçoivent les données envoyées à l'adresse 0. Des sommes de contrôle (CRC) assurent l'intégrité des différents paquets. Les esclaves ont des registres internes pour diverses fonctions. Le **tableau 1** donne un aperçu des fonctions Modbus.

Il existe trois versions du protocole Modbus :

- Modbus RTU : transmission binaire sur RS485 (EIA485).
- Modbus ASCII : transmission en texte clair sur RS485 (EIA485). C'est moins efficace que le RTU, mais lisible par l'homme. Un simple utilitaire de terminal peut envoyer les commandes.
- Modbus TCP : les commandes Modbus sont transmises par TCP/IP. C'est en général une transmission par Ethernet, mais ici, c'est du sans fil.

Le site web de Modbus [2] donne des informations plus détaillées et décrit les protocoles.

Préparation de la carte Modbus sans fil

Pour utiliser la carte sans fil avec le protocole Modbus, il faut d'abord charger le micrologiciel requis. Pour cela, retirez le module NodeMCU de la carte Modbus et connectez-le à un port USB de l'ordinateur. Ensuite ouvrez l'EDI Arduino, configuré comme décrit dans la 1ère partie de cet article. Téléchargez le micrologiciel qui transforme le module en client Modbus, depuis la page du projet Elektor [3].
Ouvrez le fichier téléchargé *OpenPLC_ESP8266_1_0_MUX_V1_1.ino* dans l'EDI Arduino (*Fichier -> Ouvrir*). L'EDI Arduino demande si le projet doit être déplacé vers le dossier des croquis. Répondez '*Oui*'. Si l'EDI crée un nouveau dossier, copiez le fichier déclaratif (*modbus.h*) dans le nouveau dossier contenant le fichier *.ino*, afin que le compilateur l'y trouve.
Pour « accéder sans fil » au module, remplacez les données d'accès du début du code source (voir **listage 1)** par les données d'accès du réseau sans fil utilisé.
Une fois cette modification effectuée, on peut charger le programme dans la carte Modbus (voir la 1ère partie). La carte fonctionne avec

Tableau 1. Aperçu des fonctions Modbus.

Description	Mode	Bits
Entrée/sortie individuelle « Bobine »	Lecture/écriture	1
« Entrées discrètes » individuelles	Lecture seulement	1
« Registres d'entrée » (analogique/numérique)	Lecture uniquement	16
Entrées/sorties « registres de stockage » (analogiques/numériques)	Lecture/écriture	16

DHCP, ce qui signifie que le routeur lui attribuera automatiquement une adresse IP. C'est vérifiable dans le micrologiciel du routeur, ou bien en observant la sortie du programme sur le moniteur série de l'EDI Arduino (à ouvrir avec l'icône de la loupe en haut à droite). Dans la boîte déroulante en bas à droite, réglez le débit de données de l'interface sur 115 200 bauds. Le **listage 2** montre une façon de sortir l'adresse IP sur le moniteur série. Pour ne pas répéter cette procédure à chaque mise sous tension de la carte Modbus, il faut lui attribuer une adresse IP statique. Tous les routeurs récents le permettent. Dans tous les cas, il faut noter l'adresse IP de la carte fournie par le programme afin de pouvoir y accéder *sans fil*. L'adresse IP n'étant émise que pendant le processus de démarrage, si aucune sortie n'apparaît, appuyez sur le bouton *reset* du module NodeMCU pour redémarrer.

Listage 1. Configuration sans fil de la carte Modbus.

```
/*********NETWORK CONFIGURATION*********/

const char *ssid = "<YOUR_SSID>";
const char *password = "<YOUR_PASSWORD>";

/*****************************************/
```

Listage 2. Sortie de l'adresse IP de la carte Modbus sans fil.

```
Connecting to Vodafone-3980
.......
WiFi connected
Server started
My IP: 192.168.0.85
```

Premier test

Pour vérifier si la carte Modbus sans fil fonctionne correctement, l'outil *EasyModbus* convient. Téléchargez-le depuis SourceForge [4]. EasyModbus fournit un serveur, un client et une bibliothèque. Pour ce test, seul le client est nécessaire. Il peut être téléchargé directement [5]. Pour utiliser cet outil, il faut installer une version de Java sur le PC. Si Java n'est pas encore installé, le logiciel gratuit OpenJDK [6] convient. Il est plébiscité par la communauté des développeurs et des créateurs et c'est un composant standard de toutes les distributions *Linux*. Dans Ubuntu, par exemple, la commande `sudo apt install openjdk-11-jdk` l'installe.

Pour lancer le client EasyModbus, entrez la commande `java -jar EasyModbusJavaClient.jar`. La **figure 1** montre la sortie du client EasyModbus (à gauche sous Linux ; à droite sous Windows). Ici sélectionnez *Modbus TCP* et entrez l'adresse IP de la carte. Laissez l'*adresse de départ* sur *1* et changez le *nombre de valeurs* : *4*. Cliquez ensuite sur *Read Discrete Inputs - FC 2* pour lire et afficher les quatre entrées numériques de la carte. Appliquez un signal à au moins une des entrées afin de voir si tout fonctionne comme prévu.

Le client EasyModbus ne peut que lire des données envoyées par la carte Modbus ; il ne peut pas en écrire sur le « bus ». Pour tester l'écriture, utilisez le programme d'exemple décrit ci-dessous. Les données effectivement transmises peuvent être vues en bas de la

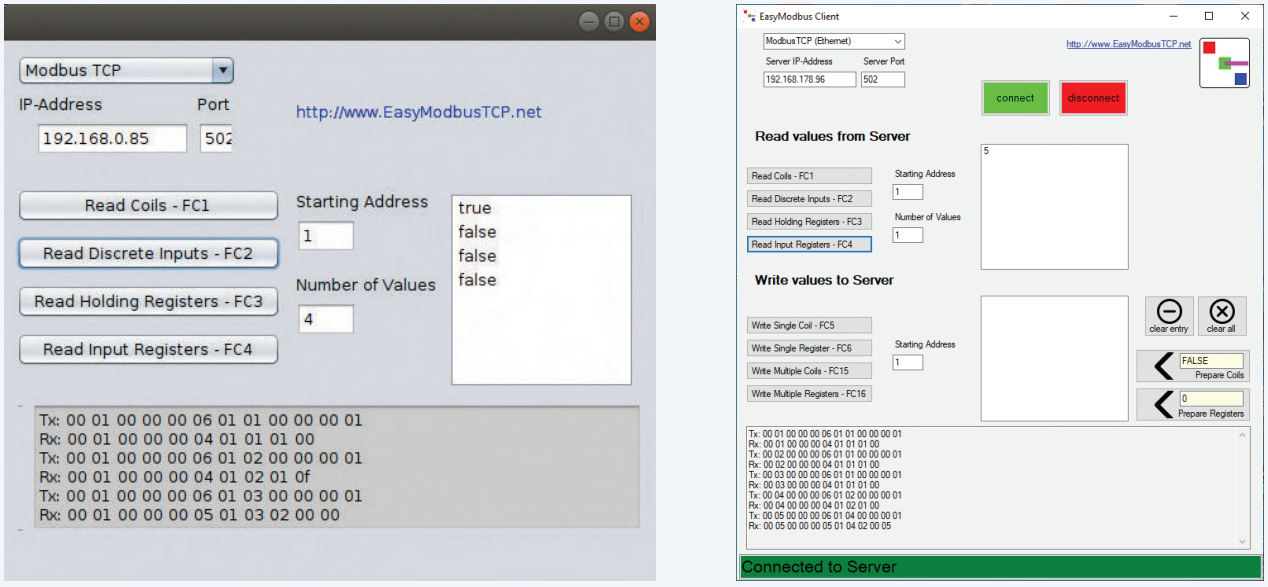


Figure 1. Le client EasyModbus en train de lire les données de la carte. La version Linux est à gauche, et la version Windows à droite.

Listage 3. Exemple de programme tor.py pour piloter la porte.

```
from pyModbusTCP.client import ModbusClient
client = ModbusClient(host="192.168.0.85",
   port=502, auto_open=True, debug=False)
while(True):
    inputs=client.read_discrete_inputs(0,4)
    end_switch_top  = inputs[0]
    end_switch_bottom = inputs[1]
    push_button_down     = inputs[2]
    push_button_up       = inputs[3]
    motor_up        = 0
    motor_down      = 1
    # request end_switch
    if(end_switch_top):
        client.write_single_coil(motor_up,False)
        print("gate open")
    if(end_switch_bottom):
        client.write_single_coil(motor_down,False)
        print("gate closed")
    # request push button
    if (push_button_up and not end_switch_top):
        client.write_single_coil(motor_up,True)
    if (push_button_down and not
            end-switch_bottom):
        client.write_single_coil(motor_down,True)
```

fenêtre du client EasyModbus. Cette fonction est très utile pour le débogage. Si le premier test est réussi, passez à la programmation de la carte en Python.

Installation de la bibliothèque Modbus

Pour rendre le test plus réaliste, nous utilisons ici la maquette de porte d'ascenseur de la 1ère partie [7]. Nous avons écrit un programme pour le PC qui scrute les positions des boutons et capteurs de fin de course par Modbus pour transmettre aussi les actions idoines aux moteurs par Modbus.

Notre programme Python a besoin d'une bibliothèque [8] afin d'accéder au Modbus. Cela n'a rien d'inhabituel ; il existe des bibliothèques Modbus pour presque tous les langages de programmation. Au besoin, on peut créer soi-même les séquences de bits des commandes et les envoyer sur le réseau, mais ce n'est pas notre propos ici.

D'abord, assurez-vous que l'installateur de paquets Python *pip* est présent, ce qui doit toujours être le cas avec des installations Python assez récentes. Pour le vérifier, utilisez `$ python -m pip -version` sous Linux ou `C:\> py -m pip -version` sous Windows.

Si l'installeur *pip* n'est pas présent, installez-le via Python avec `get-pip.py` [9] (ici n'utilisez pas le gestionnaire de paquets Linux) :

```
wget https://bootstrap.pypa.io/get-pip.py
python get-pip.py
get-pip.py
```

Ensuite, installez la bibliothèque Modbus :

```
$ python -m pip install -U pymodbusTCP
C:\> py -m pip install -U pymodbusTCP
```

Une documentation complète sur la bibliothèque et de nombreux exemples de code sont disponibles [10].

Exemple de programme

Examinons en détail le programme d'exemple du **listage 3**. La 1ère ligne importe la bibliothèque Modbus. La suivante crée un nouvel objet de connexion afin de communiquer avec un appareil Modbus spécifique. À cet effet, on peut passer plusieurs paramètres. Dans notre cas, nous passons l'adresse IP de l'appareil cible comme `hôte`. *502* est le port par défaut pour la communication Modbus TCP. Le paramètre `auto_open` détermine si la connexion doit être établie automatiquement ou manuellement. Dans notre exemple, nous avons choisi `True` pour des raisons de commodité. Pour avoir un contrôle total de la configuration de la connexion, choisissez `False`. Avec la configuration manuelle de la connexion, il y a plus d'options de contrôle et de gestion des erreurs. Si le paramètre `debug` est défini sur `True`, le logiciel affiche sur la console toutes les données transmises. C'est très utile pour le débogage.

La boucle `while`, permet l'exécution répétée du programme décrit dans la 1ère partie. Ainsi le programme tourne cycliquement, comme dans tout contrôleur industriel. La partie principale du programme est le code interne de la boucle qui lit d'abord les entrées et les convertit ensuite en variables pertinentes. Des variables pour les deux sorties numériques sont aussi définies. Cela facilite la lecture des lignes de programme suivantes.

On voit que le moteur s'arrête lorsque les fins de course sont atteintes. Un message de type texte est envoyé à la console pour que la position en cours de la porte s'affiche sur l'ordinateur. Enfin, le moteur entraîne la porte dans le sens indiqué par les boutons Haut et Bas. Le ET logique entre les capteurs de fin de course empêche d'activer les relais si la porte se trouve déjà en fin de course.

Pour lancer le programme, entrez la ligne de commande `python tor.py`. Comme d'habitude, *Ctrl-C* arrête le programme.

AdvancedHMI

Avec divers *frameworks* pour PC, il est possible de réaliser une télécommande pratique avec retour visuel, adaptable à vos besoins. Le programme *AdvancedHMI* est un logiciel libre permettant de créer des interfaces homme-machine (*IHM*) qui communiquent avec un automate ou d'autres dispositifs d'E/S. Ce logiciel diffère des autres progiciels standard dans la mesure où il permet de produire des fichiers exécutables au lieu de simples configurations interprétées par un moteur d'exécution. Il en résulte des applications très rapides et efficaces.

AdvancedHMI [11] est basé sur le *framework* .NET de Microsoft. Les applications sont générées dans *Microsoft Visual Studio Community 2019*, qui est disponible gratuitement. Cela permet de créer des IHM rudimentaires par glisser-déposer sans avoir à écrire le moindre code. Un grand nombre de développeurs utilise le *framework* .NET qui dispose de nombreux réseaux d'assistance. AdvancedHMI assure une assistance technique bien meilleure que tous les autres progiciels IHM standard réunis.

L'auteur a utilisé cette plateforme pour créer une interface utilisateur permettant de contrôler la carte industrielle. Elle indique l'état des entrées et fournit des boutons utilisables pour commuter les sorties (**fig. 2**). Les programmes .NET, tels que cette interface utilisateur pour contrôler la carte industrielle, peuvent également fonctionner sous Linux avec *Mono*. Elle a été testée avec succès sur un *Raspberry Pi*.

Prenez les commandes…

La carte Modbus sans fil permet de piloter facilement des appareils conformes au protocole Modbus à partir d'un PC ou d'un smartphone. Ce projet a une bonne valeur didactique pour ceux qui souhaitent se familiariser avec le protocole Modbus. Cependant, la conception matérielle est si robuste et tolérante aux erreurs qu'elle peut aussi s'utiliser dans les projets ESP8266 personnels pour commander et scruter des appareils industriels p. ex. des électrovannes, moteurs et autres types de capteurs ou actionneurs.

200507-B-04

Figure 2. Le logiciel AdvancedHMI produit des applications d'interface homme-machine.

Des questions, des commentaires ?

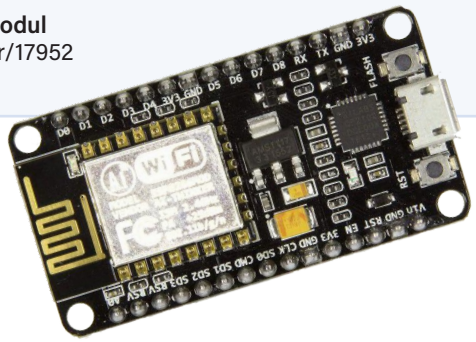
Envoyez un courriel à l'auteur (josef@bernhardt.de) ou contactez Elektor (redaction@elektor.fr).

Contributeurs

Idée, conception et texte : **Josef Bernhardt** et **Martin Mohr**
Rédaction : **Rolf Gerstendorf**
Mise en page : **Giel Dols**
Traduction : **Yves Georges**

PRODUIT

› **NodeMCU-Modul**
www.elektor.fr/17952



LIENS

[1] Modbus sans fil (partie 1), Elektor, 09-10/2021 : http://www.elektormagazine.fr/200507-04
[2] Site web Modbus : https://modbus.org/
[3] Page du projet Elektor : https://www.elektormagazine.fr/200507-B-04
[4] Site web EasyModbus : https://bit.ly/2QDIDJz
[5] Téléchargement du client EasyModbus : https://bit.ly/3d2zzFp
[6] Téléchargement d'OpenJDK : https://bit.ly/2OVK4m6
[7] Maquette de porte d'ascenseur sur YouTube : https://youtu.be/VHIBQswdA0E
[8] Bibliothèque Modbus pour Python : https://pypi.org/project/pyModbusTCP/#description
[9] get-pip.py : https://bootstrap.pypa.io/get-pip.py
[10] Exemples de bibliothèque Modbus : https://pymodbustcp.readthedocs.io/en/latest/
[11] AdvancedHMI : https://www.advancedhmi.com/index.php?main_page=page&id=14&chapter=0
[12] Plus d'infos par l'auteur : https://bit.ly/3ch1PFl

voyage dans les réseaux neuronaux (3e partie)

Les neurones pratiques

Stuart Cording (Elektor)

Jusqu'à présent, nous avons créé un réseau neuronal et compris comment il fonctionne. Nous avons même pu lui apprendre les mécanismes de la fonction OU exclusif (XOR), qui nécessite la classification de modèles d'entrée. Dans cet article, nous examinerons comment mettre en œuvre une partie d'un système de conduite autonome : reconnaître l'état de feux de circulation.

De nombreuses prédictions ont été faites concernant les voitures à conduite autonome [1], mais aucune ne s'est concrétisée. Une grande partie de la complexité provient de la tentative de comprendre l'intention des autres usagers de la route et des piétons. Cependant, le système de conduite autonome consiste surtout à classer tout ce que les nombreux capteurs et caméras « voient » autour du véhicule. Il peut s'agir de feux ou de panneaux de signalisation, mais aussi de marquage des rues ou des routes, ou encore de types de véhicules et de personnes. Maintenant que nous disposons d'un réseau neuronal fonctionnel sous la forme de notre perceptron multicouche (MLP), utilisons-le pour détecter les couleurs d'un feu de signalisation.

L'un des avantages de l'environnement Processing utilisé jusqu'à présent est la facilité avec laquelle il peut accéder aux webcams. Les exemples décrits ici fonctionneront sur à peu près n'importe quel ordinateur portable ou PC sous Windows, Linux ou macOS possédant une caméra intégrée ou externe.

Identifier votre caméra

Avant de commencer, il faut ajouter à Processing une nouvelle bibliothèque pour accéder à toutes les webcams connectées. Dans le menu, sélectionnez *Sketch -> Import Library... -> Add Library...* (**fig. 1**), ce qui ouvre la fenêtre illustrée par la **figure 2**. En vous assurant que l'onglet *Libraries* est sélectionné, entrez « *Video* » dans le champ de recherche. Dans la liste qui s'affiche, nous sélectionnons *Video | GStreamer-based video library for Processing*, puis nous cliquons sur *Install*. Comme précédemment, nous utilisons le code du dépôt GitHub préparé pour cette série d'articles [2].

Une fois la bibliothèque installée, nous pouvons exécuter le projet */trafficlight/findcamera/findcamera.pde*. Ce code demande simplement la liste des caméras disponibles connectées à l'ordinateur et permet à l'utilisateur d'en sélectionner une en saisissant un nombre entre 0

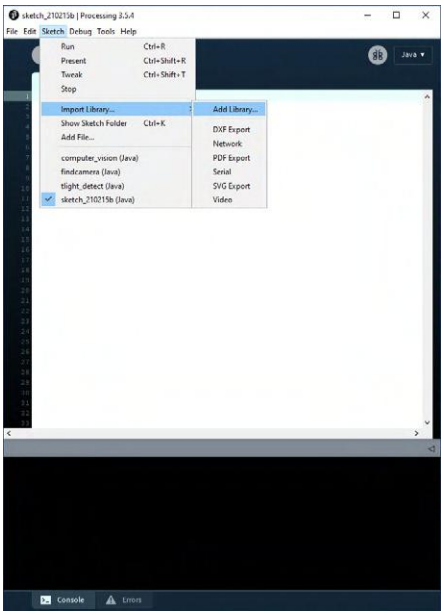


Figure 1. Ajout d'une nouvelle bibliothèque dans Processing.

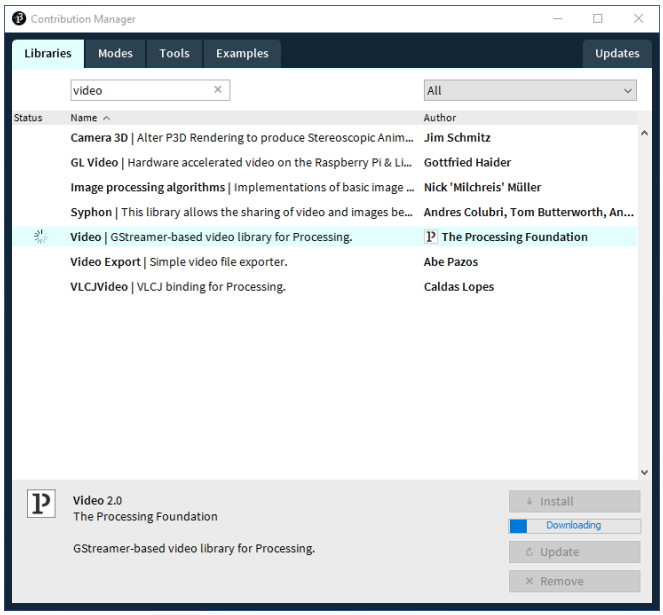


Figure 2. Recherche et installation de la bibliothèque Video.

et 9. Assurez-vous que la petite fenêtre *findcamera* soit active (cliquez à l'intérieur de la fenêtre) avant d'appuyer sur un nombre. Si la fenêtre de l'EDI Processing est active, vous taperez simplement un nombre quelque part dans le code source.

Une fois qu'une source de caméra a été sélectionnée, sa sortie est affichée en basse résolution (320×240 pixels) dans la fenêtre (**fig. 3**). La console texte fournit également le code source nécessaire à la sélection de votre caméra pour les deux prochains projets (**fig. 4**).

Enfin, nous avons besoin d'un feu de signalisation. Comme il est peu probable que vous en ayez un sous la main, il en a été préparé un dans *trafficlight/resources* avec différents formats de fichiers. Il suffit d'en imprimer un et de le garder à portée de main.

Lorsque vous arrêtez des projets Processing qui utilisent la caméra, vous pouvez voir le message « *WARNING: no real random source present !* » (avertissement : aucune source aléatoire réelle n'est présente). Il semble qu'il s'agisse d'un bogue lié à l'utilisation de la bibliothèque *Video* mais sans aucun impact sur la fonctionnalité du code.

Comment les caméras voient-elles ?

L'un des aspects les plus importants de l'utilisation des réseaux neuronaux consiste peut-être à comprendre comment un ordinateur peut interpréter les données entrantes. Dans le cas d'une entrée de type caméra, l'image informatique s'affiche en mélangeant les trois couleurs primaires, rouge, vert et bleu. C'est ce que l'on appelle communément le format RVB. Ces trois valeurs varient entre 0 et 255 (ou 0x00 et 0xFF en hexadécimal). Si nous dirigeons la caméra vers quelque chose de bleu, nous nous attendons à ce que la valeur B soit relativement élevée et les autres valeurs assez faibles. Si nous la dirigeons vers quelque chose de jaune, les valeurs R et V devraient être élevées, car, ensemble, le rouge et le vert forment le jaune. Pour avoir une meilleure idée du mélange additif des couleurs, le projet */trafficlights/addititive/additive.pde* peut être intéressant (**fig. 5**).

Sachant cela, il semble logique de déterminer comment la caméra « voit » les couleurs de notre feu de signalisation et de noter les valeurs RVB qu'elle indique. Nous pouvons ensuite apprendre à notre réseau neuronal les trois couleurs des feux de signalisation afin qu'il puisse les classer en rouge, orange et vert.

Détermination des valeurs RVB

Découvrons le projet *trafficlight/computer_vision/computer_vision.pde*. Il affiche la sortie de la caméra sélectionnée à côté des valeurs RVB qu'elle indique. Avec ce projet, nous pouvons enregistrer les valeurs RVB que la caméra perçoit. Avant de commencer le code, n'oubliez pas de coller la ligne de code que vous avez déterminée dans *findcamera.pd* à la ligne 22 de ce projet. Cela garantit l'utilisation de la caméra que vous avez choisie.

En pointant la caméra vers le feu rouge (le feu doit se trouver approximativement dans le cercle en pointillés), notez la couleur vue (en haut) et les valeurs RVB qui la définissent (**fig. 6**). Les valeurs RVB sont en fait la moyenne de tous les pixels capturés par la caméra dans le carré rouge du centre. Bien

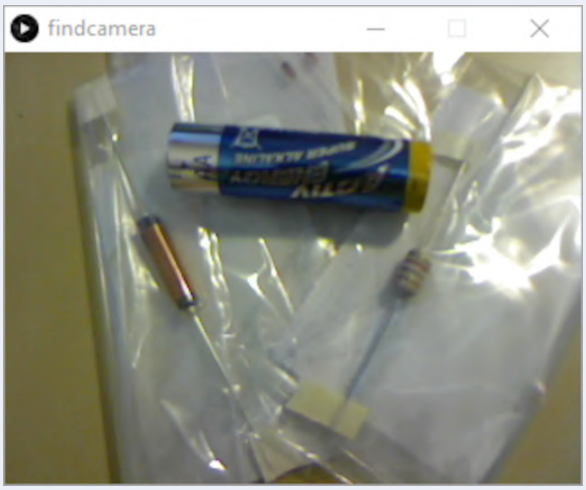


Figure 3. Exemple de sortie de findcamera.pde.

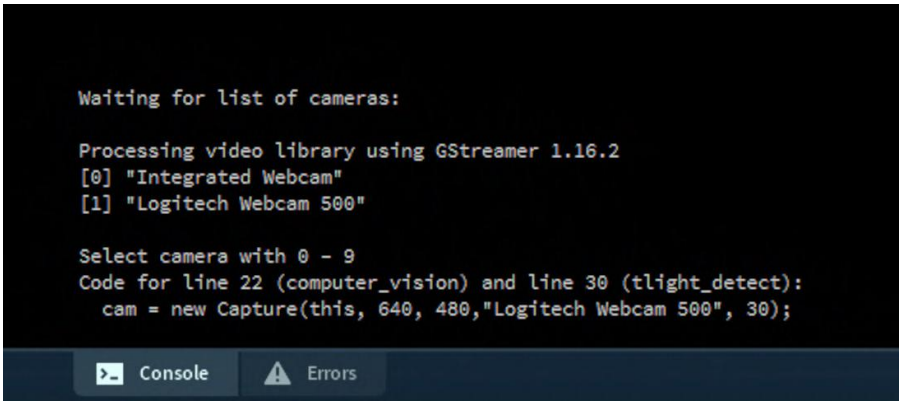


Figure 4. La sortie console de findcamera.pde fournit le code d'initialisation pour que les autres projets utilisent la caméra souhaitée.

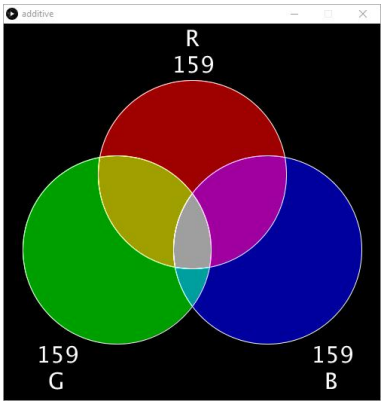


Figure 5. additive.pde montre le mélange additif de couleurs.

qu'il s'agisse d'une moyenne, les valeurs RVB varient rapidement vers le haut et vers le bas. Pour capturer une seule valeur, appuyez sur la lettre 'p' de votre clavier. Le code s'arrête alors sur un seul ensemble de valeurs. Veillez à cliquer dans cette fenêtre avant d'appuyer sur la lettre 'p' pour l'activer. En appuyant sur 's', vous redémarrez la capture de la caméra et la génération RVB.

L'impression des feux de signalisation utilisée ici a été plastifiée et présente donc quelques reflets par endroits. Si vous avez imprimé l'image des feux à l'aide d'une imprimante laser, il se peut que la surface soit légèrement réfléchissante. Dans ce cas, bien que la caméra soit dirigée vers le feu rouge, la « couleur perçue » peut être rosâtre, voire même proche du blanc. Ce qui n'est évidemment pas très utile pour le réseau neuronal. Il doit connaître la couleur « feu rouge » dans des conditions optimales. Pour cela, déplacez la caméra ou modifiez l'éclairage jusqu'à obtenir une valeur RVB représentant la couleur de manière optimale.

Cela soulève également une autre question concernant la précision. Les conducteurs savent à quel point il est difficile de discerner le feu de signalisation actif lorsque le soleil nous éblouit ou se réfléchit dans les lampes. Si nous, les humains, ne pouvons pas distinguer quelle lampe est allumée, comment un réseau neuronal peut-il le faire ? La réponse est simple : il ne peut pas le faire. Si nous avons besoin d'un algorithme plus robuste, nous devons l'entraîner avec des données de « mauvaise visibilité ». Il peut également être nécessaire de fournir une autre entrée, notamment l'endroit d'où provient la lumière du soleil, afin que le réseau neuronal sache quand la visibilité est mauvaise et qu'il utilise alors l'ensemble de données de mauvaise visibilité. Enfin, nous pourrions améliorer les données entrantes de la caméra, par exemple en ajoutant une image infrarouge du feu de circulation (indiquant les lampes chaudes) ou tout autre filtrage intelligent. Mais ne désespérez pas ! Nous allons ajouter une certaine robustesse à nos données d'entraînement pour gérer les variations de couleur des feux de circulation.

À ce stade, l'essentiel est de collecter les données RVB pour les feux rouge, orange et vert en utilisant votre caméra et vos conditions d'éclairage. Les données recueillies à l'aide de l'installation de l'auteur sont présentées dans le **tableau 1**.

Tableau 1. Valeurs RVB capturées pour les trois couleurs d'un feu de signalisation.

Couleur du feu	R	V	B
rouge	220	56	8
ambre	216	130	11
vert	123	150	128

Figure 6. computer_vision.pde crée les valeurs RVB « vues » par la caméra pour chaque couleur du feu de signalisation.

Figure 7. Réponse de tlight_detect.pde avant d'apprendre des couleurs.

Détection des feux de circulation

Armés de nos données, nous pouvons maintenant entraîner le réseau neuronal à repérer les couleurs de notre feu de circulation. Pour cette étape finale, nous aurons besoin du projet *trafficlight/tlight_detect/tlight_detect.pde* ouvert dans Processing. Commencez par vous assurer que le code d'initialisation correct de la caméra est collé dans la ligne 38 (depuis *find_camera.pde*).

Ce projet utilise un MLP à trois nœuds d'entrée, six nœuds cachés et quatre nœuds de sortie (3/6/4). Les trois entrées correspondent aux trois couleurs R, V et B. Trois des quatre sorties servent à classer les couleurs « rouge », « ambre » et « vert » du feu de signalisation. La quatrième sera utilisée ultérieurement. L'utilisation de six nœuds cachés a été choisie arbitrairement comme étant « suffisante » pour gérer la tâche de classification. Nous verrons que cette configuration fonctionne et, comme précédemment, les lecteurs sont encouragés à expérimenter avec un nombre plus ou moins grand de nœuds cachés.

Si vous exécutez le code « tel quel », le réseau neuronal fonctionne sans aucun apprentissage. Les résultats (**fig. 7**) montrent que toute couleur est classée en fonction de celles définies pour leur détection par le réseau.

L'apprentissage du réseau neuronal a lieu autour de la ligne 51. Supprimez les signes 'commentaire' des trois premières méthodes et ajoutez les valeurs RVB acquises précédemment. Les trois méthodes utilisées pour définir le rouge, l'ambre et le vert sont les suivantes :

```
teachRed(159, 65, 37);
teachAmber(213, 141, 40);
teachGreen(128, 152, 130);
```

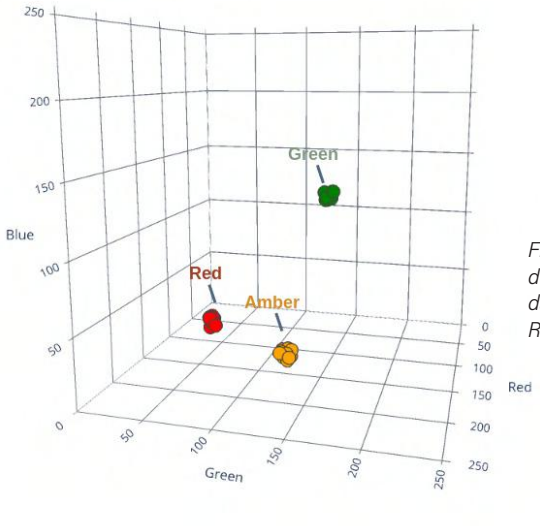


Figure 8. Les trois couleurs des feux de signalisation dans l'espace de couleurs RVB après randomisation.

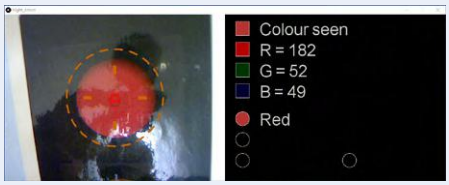


Figure 9. tlight_detect.pde après apprentissage des couleurs rouge, ambre et vert.

Chaque fois que ces fonctions sont appelées, le réseau est entraîné à classer cette combinaison RVB comme la couleur associée (**fig. 8**). Pour tenir compte de la variation de l'éclairage et des modifications automatiques du réglage de l'exposition par la caméra, une petite variation (±4) des valeurs RVB est appliquée à l'aide de la fonction `randomise()` (ligne 336). Là encore, vous pouvez expérimenter l'efficacité de cette approche et la quantité de randomisation appliquée.

L'entraînement du réseau est rapide par rapport aux projets précédents, car nous n'écrivons plus les valeurs d'erreur dans un fichier pendant l'apprentissage. Il suffit de lancer le projet. Quelques secondes après, le réseau neuronal commence à évaluer la couleur dans le carré rouge de la fenêtre de la caméra (**fig. 9**).

Le feu de circulation regardé par la caméra est déterminé à partir de la ligne 156. Les couleurs RVB capturées sont appliquées aux entrées du MLP et la sortie du réseau est calculée :

```
network.setInputNode(0, (float) r
   / 255.0);
network.setInputNode(1, (float) g
   / 255.0);
network.setInputNode(2, (float) b
   / 255.0);
network.calculateOutput();
```

La décision sur la couleur vue est ensuite prise à partir de la ligne 171. La sortie de chaque nœud de sortie est évaluée. Si la certitude de classification est supérieure à 90 % (0,90), la couleur vue est affichée dans un cercle avec le classificateur « Rouge », « Ambre » ou « Vert ».

```
// Si la probabilité de la couleur
   « Red » > 90 %...
if (network.getOutputNode(0) >
    0.90) {
  fill(200, 200, 200);
  // ...écrire « Red »…
  text(«Red», 640+(100), 320);
  // ...et fixer la couleur à
   celle perçue.
  fill(r, g, b);
} else {
  // Sinon, fixer la couleur noire
  fill(0, 0, 0);
}
// Appliquer la couleur perçue
   dans un cercle
ellipse(640+(50), 300, 40, 40);
```

Vous pouvez expérimenter la précision du MLP en pointant la caméra sur chaque feu de signalisation, sous différents angles et dans des conditions d'éclairage diverses. Essayez également de pointer la caméra sur des objets de votre voisinage qui, selon vous, se rapprochent des couleurs que le MLP a apprises.

Vous remarquerez peut-être la classification erronée d'une large gamme de couleurs comme couleur connue. Dans l'exemple de l'auteur, la classification « vert » est également donnée pour l'environnement du feu de signalisation, le cadre et l'arrière-plan de l'image (**fig. 10**).

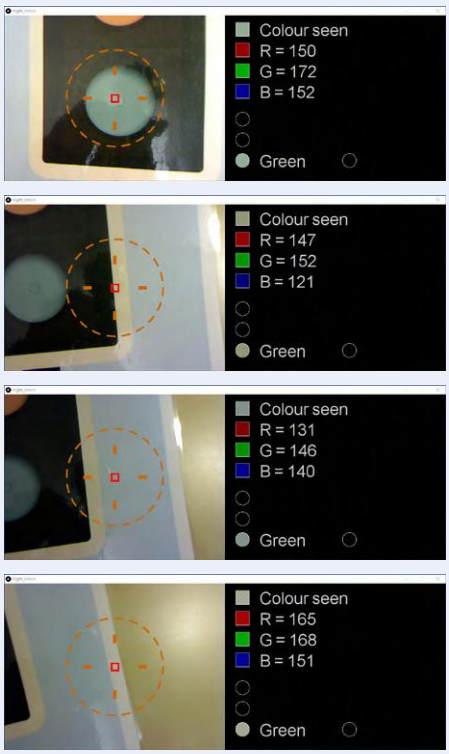


Figure 10. Classification erronée d'autres couleurs dans la catégorie « vert ».

Amélioration de la classification des réseaux neuronaux

D'après les valeurs RVB affichées, il est clair que les couleurs transmises au MLP sont raisonnablement proches de la classification souhaitée pour le « vert ». Il existe plusieurs façons de resserrer la classification. La première serait de relever la barre des 90 % pour une classification précise, avec une valeur plus élevée. Une autre approche pourrait être d'augmenter le nombre d'époques d'apprentissage. La dernière consiste à repenser la mise en œuvre de la classification.

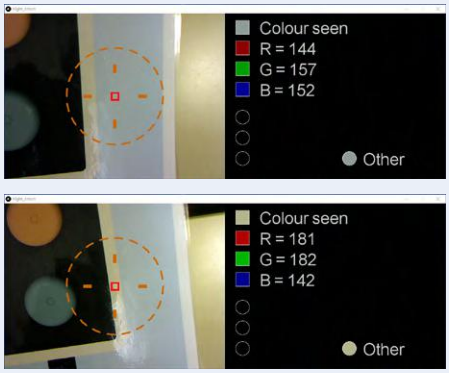


Figure 11. Classification erronée d'autres couleurs dans la catégorie « vert » résolue à l'aide du nœud de sortie « autre ».

Tableau 2. Valeurs RVB capturées pour les couleurs « indésirables ».

Objet	R	G	B
Entourage sombre du feu de signalisation	76	72	35
Bordure blanche	175	167	138
Fond bleu	152	167	161

Jusqu'ici, nous nous sommes concentrés sur ce que nous voulons classer positivement. Cependant, il peut parfois être utile d'apprendre au réseau neuronal ce qui n'appartient pas aux modèles que nous recherchons. Pour l'essentiel, nous pouvons dire : « Voici les trois éléments que nous recherchons, mais voici des éléments similaires que nous ne recherchons absolument pas ». C'est là que notre quatrième nœud de sortie entre en jeu.

En revenant une fois de plus au projet *computer_vision.pde*, nous pouvons capturer les valeurs RVB des couleurs que nous voulons classer comme « autres ». Par exemple, les valeurs RVB de l'entourage sombre du feu de circulation, de la bordure blanche et du fond bleu ont été collectées, avec les résultats présentés dans le **tableau 2**.

Dans le fichier *tlight_detect.pde*, ces valeurs peuvent être apprises comme d'autres couleurs indésirables à l'aide de la méthode `teachOther()`. Il suffit de décommenter le code autour de la ligne 60 comme suit et d'ajouter vos valeurs RVB :

```
teachOther(76, 72, 35);
teachOther(175, 167, 138);
teachOther(152, 167, 161);
```

Le fait de relancer le projet entraîne une amélioration notable. La zone autour des feux de signalisation (entourage, cadre, arrière-plan) est classée dans la catégorie « autre » (*other*) au lieu de « vert » (**fig. 11**).

Et après ?

Dans cet article, nous avons vu un réseau neuronal résoudre un problème de classification du monde réel. Nous avons également appris qu'il peut être utile d'apprendre à la fois la classification souhaitée et les données de classification non souhaitées.

Pourquoi ne pas essayer d'exploiter cet exemple pour explorer les propositions suivantes ?

- Dans quelle mesure le MLP peut-il être suffisamment « robuste » pour prendre en compte l'angle de la caméra et les changements d'exposition ? Est-il préférable de randomiser davantage les données d'apprentissage ou d'élever le niveau de classification (> 90 %) ?
- La précision s'améliore-t-elle si l'on augmente le nombre de nœuds de sortie de type « autre » et que l'on attribue à chacun une couleur indésirable ?
- Quel est l'impact sur le système de la réduction ou de l'augmentation du nombre de nœuds cachés ?
- Une quatrième entrée pour la « luminosité générale » permettrait-elle d'améliorer la précision de la reconnaissance dans des conditions d'éclairage variables ?

S'il est formidable de faire fonctionner des réseaux neuronaux sur des ordinateurs portables et des PC puissants, beaucoup d'entre nous souhaiteraient disposer d'une telle capacité sur les processeurs plus petits comme ceux d'une carte Arduino. Dans le dernier article de cette série, nous utiliserons un capteur RVB et un Arduino pour mettre en œuvre un réseau neuronal de détection des couleurs. Peut-être sera-t-il la base de votre prochain projet Arduino !

210160-C-04

Des questions, des commentaires ?

Envoyez un courriel à l'auteur (stuart.cording@elektor.com).

Contributeurs

Idée, texte et illustrations : **Stuart Cording**
Rédaction : **Jens Nickel, C. J. Abate**
Illustrations : **Patrick Wielders**
Mise en page : **Harmen Heida**
Traduction : **Pascal Godart**

PRODUITS

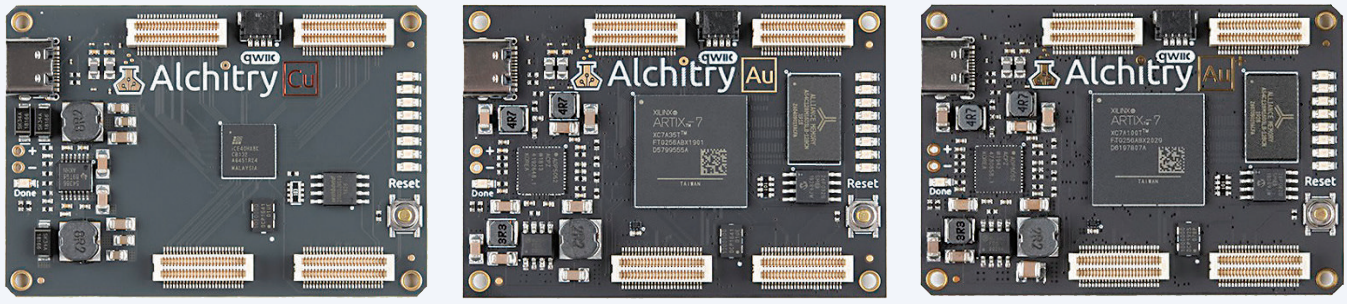
- **Livre en anglais, « Artificial Intelligence » de B. van Dam**
  www.elektor.fr/18090
- **Kit AIY Vision pour Raspberry Pi de Google**
  www.elektor.fr/19365
- **Caméra AI avec boîtier en silicone de HuskyLens**
  www.elektor.fr/19248

LIENS

[1] M. Anderson, « Surprise! 2020 Is Not the Year for Self-Driving Cars », IEEE Spectrum, April 2020 : http://bit.ly/2ZSwm5f
[2] Dépôt GitHub – réseau neuronal simple : https://bit.ly/2ZHLv9p
[3] H. Zhang et al., « StackGAN: Text to Photo-realistic Image Synthesis with Stacked Generative Adversarial Networks », CVF, décembre 2017 : https://bit.ly/3qZccCs

dans l'intimité d'un processeur *open source*

Extrait : résultats des FPGA Lattice et Xilinx



Cartes de développement FPGA d'Alchitry citées dans cet article. De gauche à droite : Cu - Au - Au+.

Monte Dalrymple (États-Unis)

Cet article est un extrait de l'ouvrage de *Monte Dalrymple*, « Inside an Open-Source Processor », publié en anglais par Elektor. Les électroniciens ont adopté la technique FPGA combinée au langage (V)HDL. De plus, avec l'architecture RISC-V, l'*open source* accède aux applications professionnelles. Cet article expose des comparaisons point par point et des résultats obtenus à partir d'un exemple d'application exécuté sur les cartes FPGA *Lattice* et *Xilinx* de la série *Alchitry* maintenant disponible chez Elektor. Mais d'abord, quelques conseils.

Note de l'éditeur : cet article est un extrait du livre *Inside an Open-Source Processor — an Introduction to RISC-V* formaté et légèrement modifié pour correspondre aux normes éditoriales et à la mise en page du magazine *Elektor*. Puisque cet article est extrait d'une publication plus vaste, certains termes peuvent faire référence à des passages du livre d'origine situés ailleurs. L'auteur et l'éditeur ont fait de leur mieux pour l'éviter et seront heureux de répondre aux questions – Pour les contacter, voir l'encadré « **Des questions, des commentaires ?** ».

Jusque-là, mettre en œuvre un projet enchaînant des outils FPGA était complexe, mais avec la maturité de la technique, c'est devenu bien plus simple. Pour l'exemple présenté ici (une horloge sur 24 h décrite ailleurs dans le livre), la majorité du travail est déjà accomplie.

Il faut quand même savoir que les gros projets, ou les projets exploitant au mieux le potentiel FPGA, seront à coup sûr bien plus complexes.

Dans le cas de Lattice et Xilinx, outre la sélection de la cible FPGA appropriée pour le projet, seuls deux ou trois fichiers devront être liés dans les outils. Pour cette approche simplifiée, tous les fichiers du projet doivent être dans le même dossier. Comme indiqué

ci-dessus, les lecteurs sont fortement encouragés à consulter les tutoriels Alchitry pour l'installation du logiciel FPGA et la configuration du projet car seul l'essentiel sera abordé ici.

Conseils Lattice (Alchitry Cu)

Le FPGA utilisé sur cette carte nécessite le logiciel Lattice *iCEcube2*. Il s'agit d'un produit mature sans fioritures. Voici la liste des points à retenir :

1. S'assurer de configurer le projet avec les options correctes. L'Alchitry Cu utilise le FPGA *HX8K* de la famille *iCE40*, en boîtier CB132. Tous les bancs d'E/S s'alimentent en 3,3 V.
2. Ne saisir que le fichier *yrv_alchitry.v* comme fichier de conception : tous les autres fichiers seront automatiquement inclus dans le bon ordre.
3. Pour l'outil de synthèse, veiller à sélectionner Lattice LSE Synthesis. Aucun fichier de contraintes n'est requis pour la synthèse logique.
4. Une fois la synthèse terminée, les fichiers de contrainte de broches *yrv_alchitry.pcf* et de synchronisation *timing.sdc* doivent être liés (*linked*) avant d'exécuter le reste des outils.

Des avertissements en nombre apparaîtront dans les journaux, mais aucun d'eux ne devrait nécessiter d'action.

Conseils Xilinx (Alchitry Au et Au+)

Les FPGA utilisés sur ces cartes nécessitent le logiciel *Vivado* de Xilinx. Il s'agit d'un produit complexe, et ce projet n'utilisera qu'un petit sous-ensemble des capacités de ce logiciel. Voici la liste des points à retenir :

1. Comme fichier de conception, ne saisir que le fichier *yrv_alchitry.v*.
2. Comme fichier de contraintes ne saisir que le fichier *yrv_alchitry.xdc*. Ce fichier contient les contraintes de broches, de timing ainsi que les spécifications de tension de programmation.
3. L'option de FPGA correcte est xc7a35tftg256-1 pour la carte Au et xc7a100tftg256-1 pour la carte Au+.

Des avertissements en nombre encore plus grand apparaîtront dans les journaux, mais là aussi aucun ne devrait nécessiter d'action.

Résultats FPGA

Ce projet-exemple a été mis en œuvre sur les trois cartes-cibles de développement FPGA. Les résultats individuels sont analysés ci-dessous. Il est intéressant de commencer par une comparaison point par point de certains des résultats. Le **tableau 1** montre la vitesse d'horloge rapportée pour le diviseur de 100 MHz.
Le diviseur de 100 MHz a été conçu pour une vitesse maximale avec un seul niveau de logique entre les flip-flops. Cela signifie que ce résultat devrait représenter les capacités de la technique.

Si la vitesse annoncée pour la puce de Lattice est vraie, c'est impressionnant. Ce qui est le plus intéressant, c'est que la vitesse

Tableau 1. Résultats FPGA.

Diviseur de 100 MHz	Alchrity Cu	Alchrity Au	Alchrity Au+
Vitesse (à la synthèse)	313,9 MHz	114,1 MHz	114,2 MHz
Vitesse (au placement)	542,3 MHz	–	–
Vitesse (finale)	625,4 MHz	111,9 MHz	111,9 MHz

Tableau 2. Résultats FPGA.

Attribut FPGA	Alchrity Cu	Alchrity Au	Alchrity Au+
Éléments Logiques	7 680	20 800	63 400
É.L. utilisés	4 646	2 189	2 185
É.L. utilisés (%)	60,6%	10,5%	3,5 %
Flip-flop	1 331	1 377	1 377
Vitesse (synthèse)	11,8 MHz	28,1 MHz	27,7 MHz
Vitesse (placement)	25,8 MHz	-	-
Vitesse (finale)	30,8 MHz	34,4 MHz	34,2 MHz

indiquée s'améliore et devient sans doute plus précise, en passant de la synthèse logique au placement logique puis au résultat final du routage. Même s'il vaut mieux que les performances finales soient meilleures que l'estimation initiale, un écart d'un facteur deux n'est pas idéal, car il pourrait motiver l'abandon d'un projet jugé à tort irréalisable.

Les outils Xilinx ne donnent pas d'estimation distincte de la vitesse au niveau du placement logique, mais le chiffre de la synthèse logique est quasi identique à celui du résultat final. Du point de vue d'un concepteur, c'est préférable.

Le **tableau 2** indique les ressources requises pour le projet ainsi que la fréquence d'horloge max. pour le CPU lui-même. Ces résultats ne tiennent pas compte des compteurs à 64 bits inclus dans l'*architecture RISC-V Privileged*.
La 2e ligne du tableau montre pourquoi la comparaison des résultats des FPGA peut être compliquée. D'après le tableau, l'implémentation de Xilinx nécessite moins de la moitié des ressources requises par l'implémentation de Lattice. Mais un élément logique Xilinx

Listage 1. Extrait de rapport de Lattice.

```
Total Logic Cells: 4646/7680
       Combinational Logic Cells: 3315     out of   7680     43.1641%
       Sequential Logic Cells:    1331     out of   7680     17.3307%
       Logic Tiles:               847      out of   960      88.2292%
   Registers:
       Logic Registers:           1331     out of   7680     17.3307%
       IO Registers:              0        out of   1280     0
   Block RAMs:                    12       out of   32       37.5%
   Warm Boots:                    0        out of   1        0%
   Pins:
       Input Pins:                36       out of   95       37.8947%
       Output Pins:               50       out of   95       52.6316%
       InOut Pins:                0        out of   95       0%
   Global Buffers:                2        out of   8        25%
   PLLs:                          0        out of   2        0%
```

contient beaucoup plus de logique qu'un élément logique Lattice. En outre, la conception Xilinx utilise des blocs DSP dédiés que la puce Lattice ne possède pas.

Le pourcentage d'éléments logiques utilisés révèle la disparité de taille entre les trois FPGA utilisés ici. Un peu plus de la moitié de la puce Lattice est nécessaire pour la conception *yrv_mcu*, mais environ un 10e de celle de Xilinx de milieu de gamme est utilisé et moins d'un 20e du grand Xilinx est nécessaire. Cette utilisation est conforme au coût des différentes cartes de développement.

Comme prévu, le nombre de flip-flops requis pour la conception est similaire sur les trois FPGA, et le nombre exact est plus une fonction de l'outil de synthèse logique qu'autre chose. Les outils de synthèse logique répliquent les flip-flops pour augmenter les performances ou simplifier le routage.

Les vitesses montrent que les trois FPGA fournissent à peu près les mêmes performances, c'est un peu surprenant car le FPGA de Lattice ne contient pas les blocs DSP spécialisés utilisés dans l'implémentation de Xilinx.

Alchitry Cu en détail

Extrait directement du rapport de mise en œuvre de Lattice, le **listage 1** montre les détails d'utilisation des ressources. Extraite d'une capture d'écran de l'outil *Lattice floorplanner*, la **figure 1** montre l'utilisation des éléments logiques tels que répartis sur le FPGA. Cette distribution est assez uniforme sur l'ensemble, même si l'utilisation globale n'est que de 60 % environ.

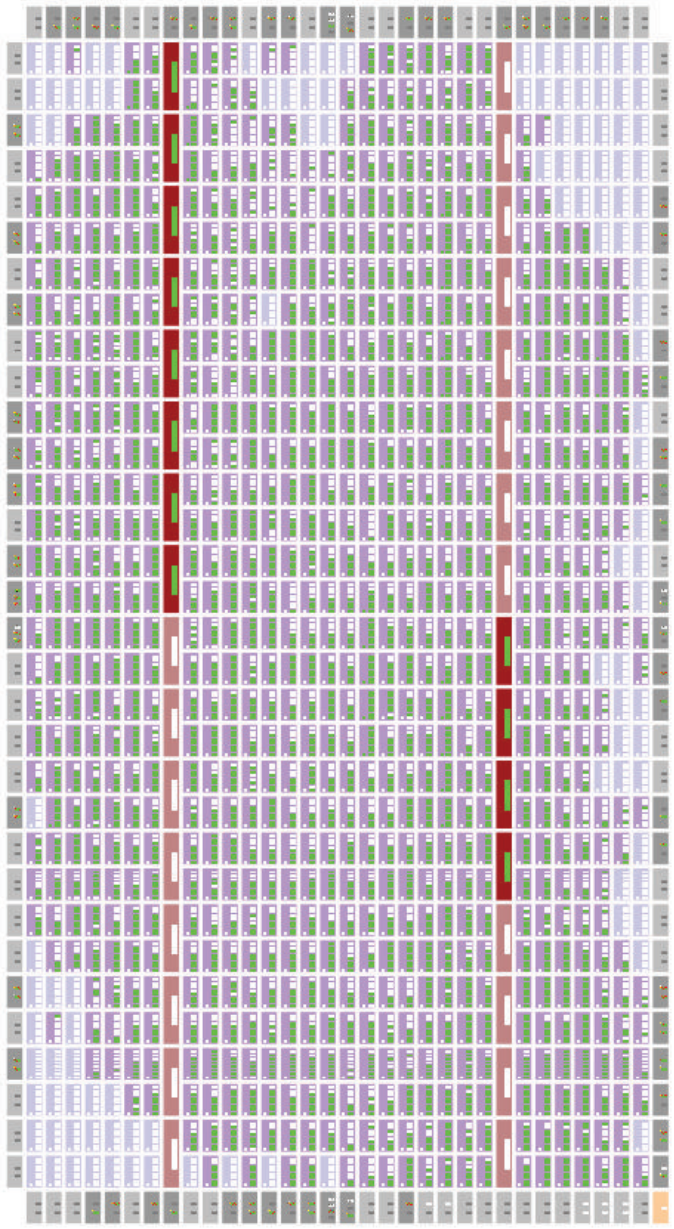


Figure 1. Vue du Floorplanner de Lattice (Alchitry Cu).

Alchitry Au en détail

Extrait directement du rapport de mise en œuvre de Xilinx, le **listage 2** montre les détails d'utilisation des ressources. Le rapport complet de Xilinx est beaucoup plus volumineux et couvre tout le matériel dédié disponible dans le FPGA.

La **figure 2** vient d'une capture d'écran de l'outil Xilinx floorplanner qui montre l'utilisation des éléments logiques répartis sur le FPGA, ainsi que la manière dont les blocs logiques dédiés y sont

Listage 2. Extrait de rapport de Xilinx XCA35T.

```
1. Slice Logic
--------------

+-------------------------+------+-------+-----------+-------+
|        Site Type        | Used | Fixed | Available | Util% |
+-------------------------+------+-------+-----------+-------+
| Slice LUTs              | 2189 |     0 |     20800 | 10.52 |
|   LUT as Logic          | 2189 |     0 |     20800 | 10.52 |
|   LUT as Memory         |    0 |     0 |      9600 |  0.00 |
| Slice Registers         | 1377 |     0 |     41600 |  3.31 |
|   Register as Flip Flop | 1377 |     0 |     41600 |  3.31 |
|   Register as Latch     |    0 |     0 |     41600 |  0.00 |
| F7 Muxes                |    3 |     0 |     16300 |  0.02 |
| F8 Muxes                |    0 |     0 |      8150 |  0.00 |
+-------------------------+------+-------+-----------+-------+
```

répartis. Il semble que les outils de placement débutent toujours par le coin inférieur gauche d'un FPGA. La logique semble divisée en deux parties, mais la raison n'est pas claire.

Alchitry Au+ en détail

Aussi extrait directement du rapport de mise en œuvre de Xilinx, le **listage 3** montre les détails d'utilisation des ressources. On note que les chiffres relatifs aux ressources utilisées sont presque identiques entre les deux FPGA Xilinx.

La **figure 3** vient aussi d'une capture d'écran de l'outil Floorplanner de Xilinx. Il est ici intéressant de noter que même si ce FPGA contient trois fois plus d'éléments logiques, la surface globale requise par l'exemple choisi semble être répartie sur une surface quasi identique à celle de la 1ère implémentation Xilinx.

Programmation matérielle

Avec le chargeur Alchitry, le téléchargement du *bitstream* (suite de bits) FPGA dans une carte de développement Alchitry est simple. Ce chargeur autonome est automatiquement installé avec le logiciel *Alchitry Labs* disponible auprès d'Alchitry. La **figure 4** montre l'interface utilisateur du chargeur.

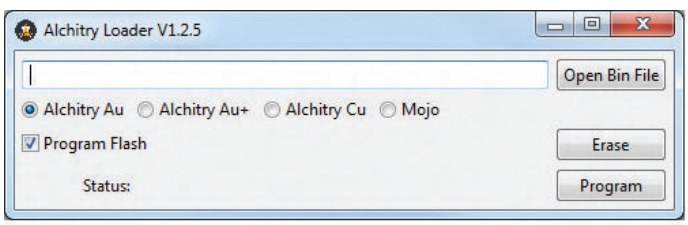


Figure 4. Vue du chargeur d'Alchitry.

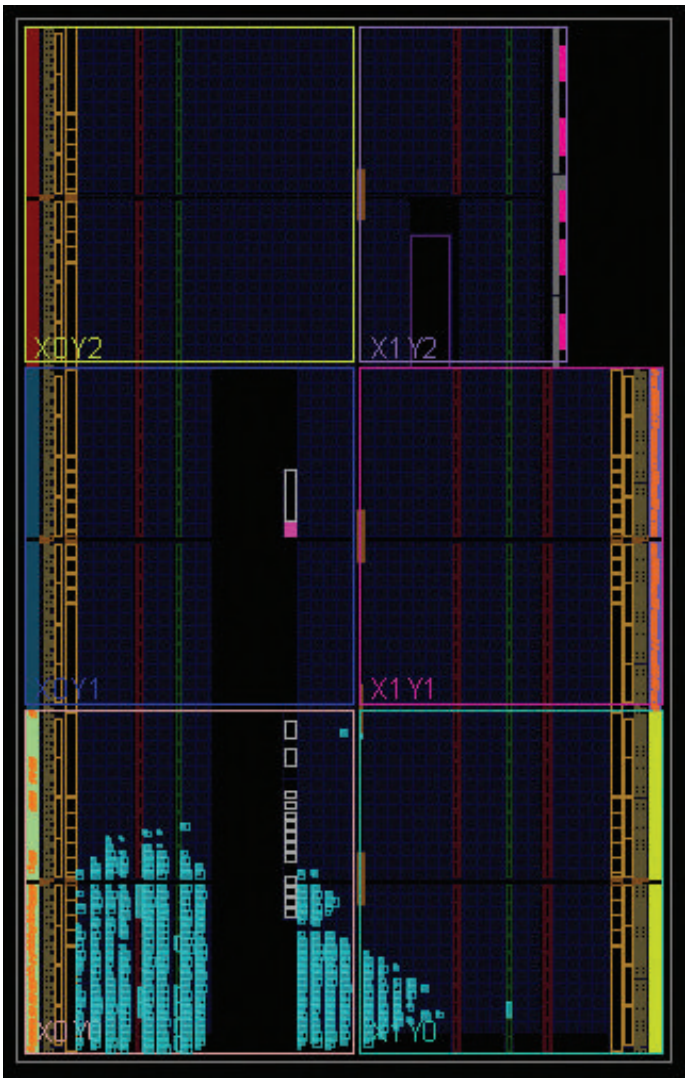


Figure 2. Vue du Floorplanner de Xilinx XCA35T (Alchitry Au).

La programmation de la mémoire Flash de la carte de développement utilisée pour charger le FPGA se résume à spécifier le fichier bitstream, sélectionner la carte cible et cliquer sur le bouton *Program*.

Par défaut, le programme recherche un fichier bitstream de type *.bin*, utilisé par défaut par le logiciel iCEcube2. Le logiciel Vivado produit un fichier bitstream de type *.bit*, il faut en tenir compte pour charger le fichier bitstream de Xilinx.

210347-04

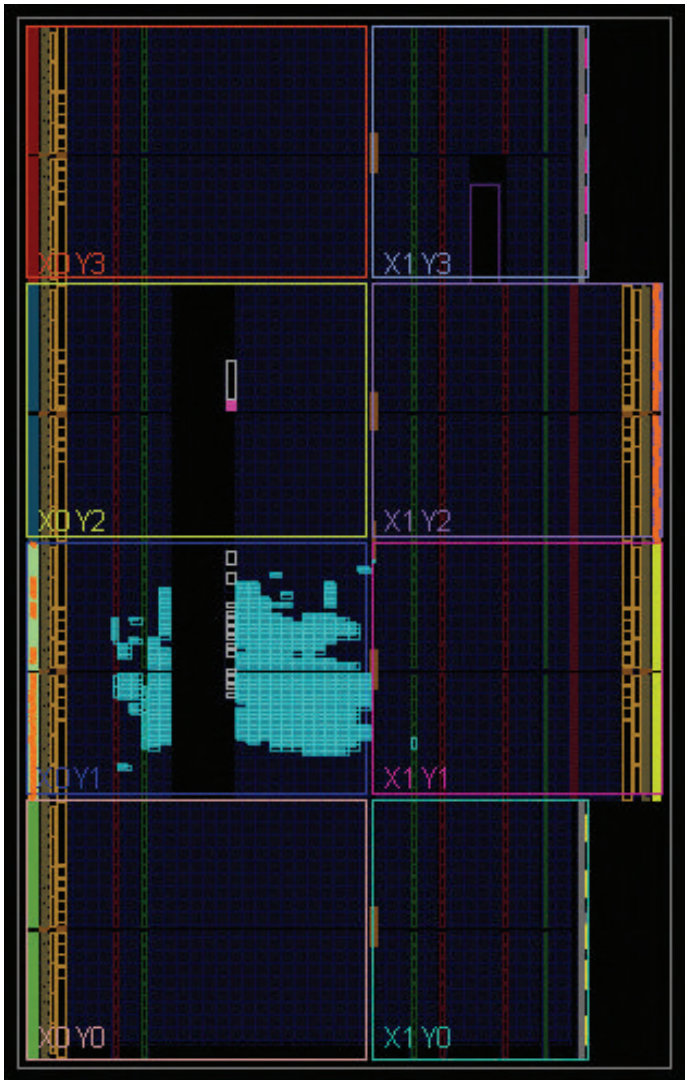


Figure 3. Vue du Floorplanner de Xilinx XCA100T (Alchitry Au+).

Listage 3. Extrait de rapport de Xilinx XCA100T.

```
1. Slice Logic
--------------

+-------------------------+------+-------+-----------+-------+
|        Site Type        | Used | Fixed | Available | Util% |
+-------------------------+------+-------+-----------+-------+
| Slice LUTs              | 2185 |     0 |     63400 |  3.45 |
|   LUT as Logic          | 2185 |     0 |     63400 |  3.45 |
|   LUT as Memory         |    0 |     0 |     19000 |  0.00 |
| Slice Registers         | 1377 |     0 |    126800 |  1.09 |
|   Register as Flip Flop | 1377 |     0 |    126800 |  1.09 |
|   Register as Latch     |    0 |     0 |    126800 |  0.00 |
| F7 Muxes                |    3 |     0 |     31700 | <0.01 |
| F8 Muxes                |    0 |     0 |     15850 |  0.00 |
+-------------------------+------+-------+-----------+-------+
```

Des questions, des commentaires ?

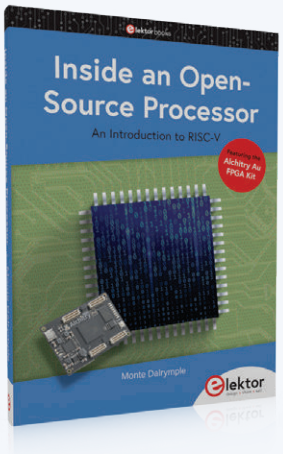
Envoyez un courriel à l'auteur (monted@systemyde.com) ou contactez Elektor (redaction@elektor.fr).

Contributeurs

Texte : **Monte Dalrymple**
Rédaction : **Jan Buiting**
Traduction : **Yves Georges**
Mise en page : **Giel Dols**

PRODUITS

- **Livre en anglais de M. Dalrymple, « Inside an Open-Source Processor », Elektor 2021**
  Version papier : www.elektor.fr/19826
  Version numérique : www.elektor.fr/19827
- **Kit FPGA Alchitry Au**
  www.elektor.fr/19638



démarrer en électronique… (10)

Toujours la bobine, mais avec ses applications pratiques dans l'audio.

Eric Bogers (Elektor)

Dans cet épisode, nous continuons avec la bobine. Peu appréciée des électroniciens amateurs, mais ô combien importante.

Circuits en série et en parallèle

Les circuits résonants (**fig. 1**) sont les circuits qui déterminent la fréquence des oscillateurs. À la fréquence dite de résonance, leur impédance est à son extrémum (minimum ou maximum). À cette fréquence, l'impédance du condensateur est égale à celle de l'inductance ; l'équation suivante s'applique :

$$X_C = \frac{1}{2 \cdot \pi \cdot f_R \cdot C} = X_L = 2 \cdot \pi \cdot f_R \cdot L$$

$$\Rightarrow f_R = \frac{1}{2 \cdot \pi \cdot \sqrt{L} \cdot C}$$

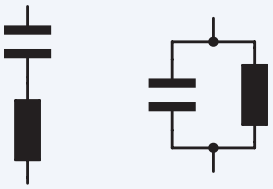


Figure 1. Circuits résonants en série et en parallèle.

Cette formule est valable pour les circuits résonants série et parallèles. La **figure 2** montre les variations d'impédance en fonction de la fréquence. Un circuit série peut être utilisé (entre autres !) dans une enceinte de haut-parleur pour atténuer une petite gamme de fréquences afin de linéariser la caractéristique globale de l'enceinte. Pour cela, dans la plupart des cas, une résistance doit être connectée en série avec ce circuit résonant série.

Filtres répartiteurs

La **figure 3** montre les courbes d'impédance et de phase d'un haut-parleur de basse (qui, par ailleurs, est monté dans une enceinte

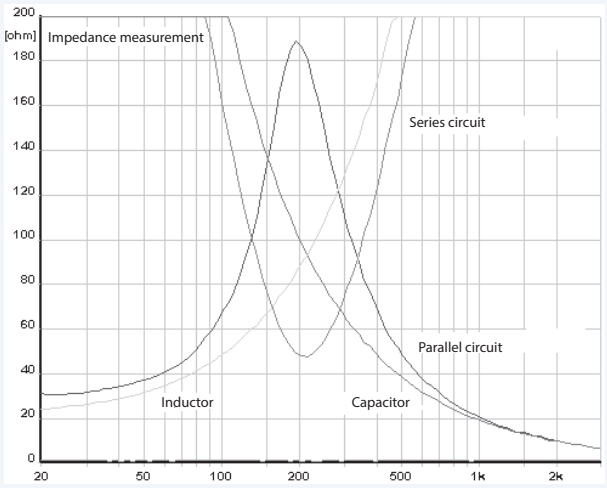


Figure 2. Courbes d'impédance des circuits résonants en série et en parallèle.

à pavillon à charge frontale, détail qu'on peut oublier). La **figure 4** montre la courbe de fréquence de ce système et la **figure 5** donne une idée de son taux de distorsion.
L'ensemble du domaine de fréquences est divisé en quatre régions. La première se situe en dessous de la fréquence de résonance – dans l'exemple de ce haut-parleur, elle va de 0 Hz à environ 30 Hz. Le rendement du haut-parleur y est franchement médiocre et le taux de distorsion très élevé, tandis que le déphasage entre le courant et la tension est compris entre 60° et 90°. Cela signifie également que la puissance dissipée dans l'amplificateur est la plus élevée. Pour ces raisons, ce haut-parleur est inutilisable dans cette gamme de fréquences : il a à la fois un son médiocre et un mauvais rendement, il consomme une puissance électrique excessive, ce qui fera surchauffer l'amplificateur (avec toutes les conséquences néfastes que cela peut entraîner).
Pour ce haut-parleur, il faut utiliser un filtre passe-haut avec une fréquence de coupure d'au moins 50 Hz. Pour la reproduction de fréquences plus basses, il faudra utiliser un autre type de haut-parleur et/ou un autre type d'enceinte.
La deuxième région se situe autour de la fréquence de résonance où l'impédance (et donc aussi le rendement) augmente de façon spectaculaire. Le plus souvent, les caissons de haut-parleurs fonctionnent plutôt bien dans cette gamme de fréquences (ce qui n'est hélas pas le cas dans cet exemple…).
Dans la troisième région, la courbe d'impédance est à peu près horizontale et le rendement est acceptable avec un petit taux de distorsion ; la courbe de phase ne donne pas non plus de raison de s'inquiéter. Dans l'exemple, cette région se situe entre 100 Hz et 1 kHz environ – c'est la gamme de fréquences pour laquelle ce haut-parleur devrait être utilisé. La courbe de distorsion montre une distorsion de second ordre relativement forte (et même des harmoniques), c'est caractéristique des systèmes à pavillon.
Enfin, dans la quatrième région, l'impédance augmente progressivement en raison de l'auto-inductance de la bobine mobile, le niveau de sortie diminue nettement et la distorsion augmente à nouveau. Ce point est la limite supérieure de la fréquence du signal applicable à ce haut-parleur. Si, avec des filtres passifs (voir plus loin), la fréquence de coupure tombe dans cette quatrième région, il faut utiliser une compensation d'impédance (ce sujet sera abordé dans un prochain article).

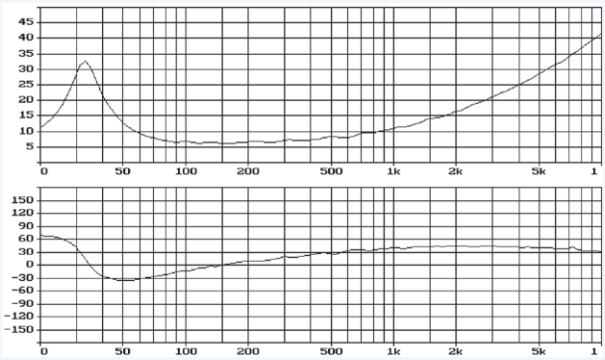


Figure 3. Courbes d'impédance et de phase.

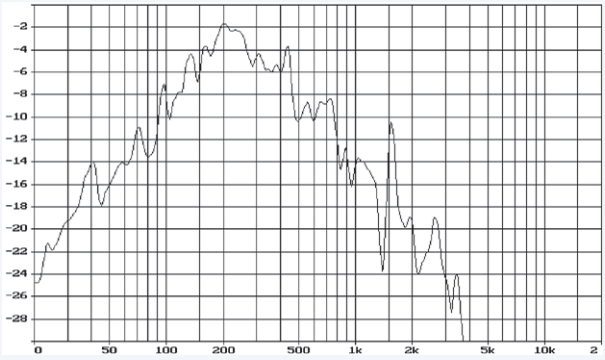


Figure 4. Courbe de fréquence.

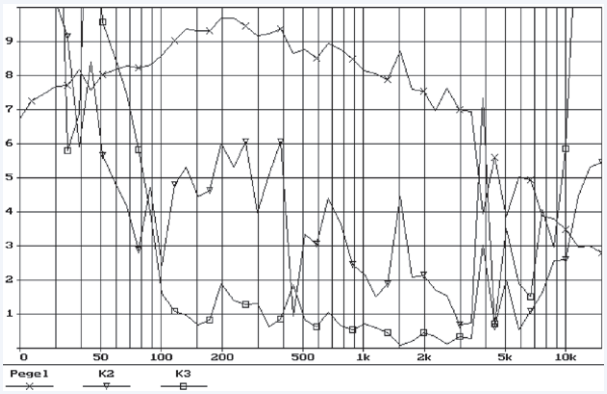


Figure 5. Distorsion.

Pourquoi des filtres répartiteurs ?

Comme nous l'avons vu dans l'exemple ci-dessus, les haut-parleurs ne peuvent bien reproduire qu'une gamme de fréquences limitée. C'est pourquoi le spectre audio doit être divisé en un certain nombre de régions, chacune d'entre elles étant ensuite envoyée à un haut-parleur conçu spécifiquement pour elle. C'est le rôle du filtre répartiteur (ou d'aiguillage), qui veille à ce que chaque haut-parleur reçoive les fréquences adéquates.
En principe, cela peut se faire de deux manières différentes : le spectre du signal peut être divisé après l'amplificateur de puissance à l'aide d'un réseau de filtres passifs (qui comprend des inductances et des condensateurs), ou bien il peut être divisé en amont selon les différentes régions, avec pour chacune son propre amplificateur de puissance et son propre haut-parleur. Dans ce cas, on parle d'un filtre répartiteur actif.

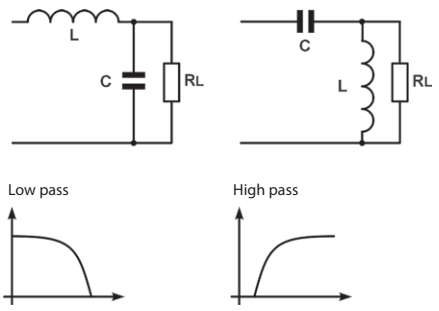


Figure 6. Filtres passifs passe-bas et passe-haut avec une pente de 12 dB/octave.

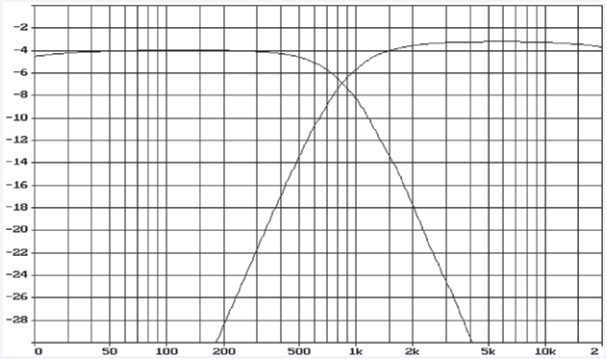


Figure 7. Courbes de fréquence d'un filtre passe-haut et d'un filtre passe-bas.

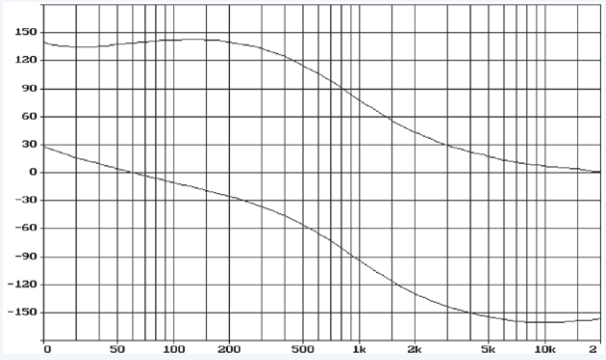


Figure 8. Courbes de phase d'un filtre passe-haut et d'un filtre passe-bas.

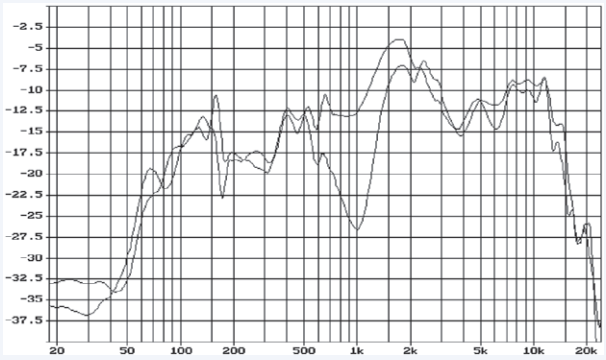


Figure 9. Filtre répartiteur correctement et incorrectement connecté.

Par ailleurs, une forme hybride est également possible : dans ce cas, les basses fréquences sont séparées du reste au moyen d'un filtre actif, et pour ce reste, on utilise un filtre passif. C'est ce qu'on appelle parfois la bi-amplification.
Nous poursuivons ci-dessous l'étude des filtres passifs ; les filtres actifs feront l'objet d'un article ultérieur.

Filtres passe-haut passifs

La **figure 6** montre les schémas d'un filtre passe-bas et d'un filtre passe-haut avec une pente de 12 dB/octave. Les pentes de 6 dB/octave, respectivement réalisables avec une inductance ou un condensateur en série, doivent être évitées dans les applications audio. La **figure 7** illustre les courbes de fréquence de ces filtres.
La fréquence de coupure d'un filtre est la fréquence à laquelle le niveau du signal de sortie est réduit de 3 dB par rapport au signal d'entrée. La **figure 8** montre qu'à la fréquence de coupure, la phase du signal de sortie est décalée de +90° (filtre passe-haut) ou de –90° (filtre passe-bas) par rapport à la phase du signal dans la bande passante.
Lorsque les haut-parleurs sont connectés avec les mêmes polarités, leurs signaux dans la région de la fréquence de coupure sont en opposition de phase et vont donc s'annuler. Il en résulte un « creux » dans la caractéristique du système, comme on peut le voir sur la courbe inférieure de la **figure 9**.
En principe, peu importe lequel des deux haut-parleurs doit avoir ses bornes inversées, mais pour garantir que deux enceintes placées l'une à côté de l'autre soient en phase, il est recommandé de s'en tenir à la « norme » internationale : le haut-parleur basse fréquence est connecté normalement et le haut-parleur haute fréquence est connecté avec une polarité inversée.
Cela dit, en pratique, nous ne pouvons pas nous fier aveuglément à la façon dont le fabricant repère les bornes de ses haut-parleurs – d'abord parce qu'il est libre de faire ce qui lui plaît, et ensuite parce qu'en raison d'un retard, on peut se retrouver avec une opposition de phase à la fréquence de coupure. Que ce soit avec des filtres actifs ou passifs, la polarité des haut-parleurs doit toujours être vérifiée dans la pratique.
Pour les valeurs des composants, la règle suivante s'applique (pour une caractéristique Butterworth) :

$$C = \frac{1}{2 \cdot \sqrt{2} \cdot \pi \cdot f \cdot Z} = \frac{0.1125}{f \cdot Z}$$

$$L = \frac{Z}{\sqrt{2} \cdot \pi \cdot f} = \frac{0.2251 \cdot Z}{f}$$

La conception des filtres répartiteurs repose sur différents critères, chacun ayant ses avantages et ses inconvénients en ce qui concerne la pente à la fréquence de coupure et le comportement impulsionnel. Pour plus de détails, nous vous renvoyons à la littérature technique sur le sujet.

La **figure 10** montre le schéma de base d'un filtre passe-bande, qui n'est en fait qu'un montage en série d'un filtre passe-bas et d'un filtre passe-haut. Les fréquences de coupure supérieure et inférieure sont ensuite calculées selon la méthode habituelle.

210564-04

La série d'articles « démarrer en électronique » est basée sur le livre « Basic Electronics Course » de Michael Ebner, publié par Elektor.

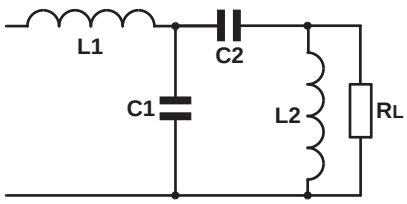


Figure 10. Filtre passe-bande.

Des questions, des commentaires ?

Contactez Elektor (redaction@elektor.fr).

Contributeurs

Idée et illustrations : **Michel Ebner**
Texte et rédaction : **Eric Bogers**
Traduction : **Helmut Müller**
Mise en page : **Giel Dols**

PRODUITS

- **B. Kainka, *Initiation à l'électronique et programmation de montages pour débutants***
  www.elektor.fr/19339
- **R. Mallard, *L'électronique pour les débutants***
  www.elektor.fr/15662

Vous souhaitez publier votre montage dans le magazine ?

Rendez-vous sur la page du labo d'Elektor : www.elektormagazine.fr/labs pour y enregistrer votre projet.

Cliquez sur « Créer un projet ». Connectez-vous (créez un compte gratuit si vous n'en avez pas encore). Remplissez les différents champs du formulaire.

Votre proposition de montage sera examinée par l'ensemble des rédacteurs du magazine. Si votre projet est retenu pour sa publication dans le magazine, un rédacteur prendra contact avec vous pour vous accompagner dans la rédaction de l'article.

Labo d'Elektor :
www.elektormagazine.fr/labs

créer > partager > vendre

projet 2.0

corrections, mises à jour et courriers des lecteurs

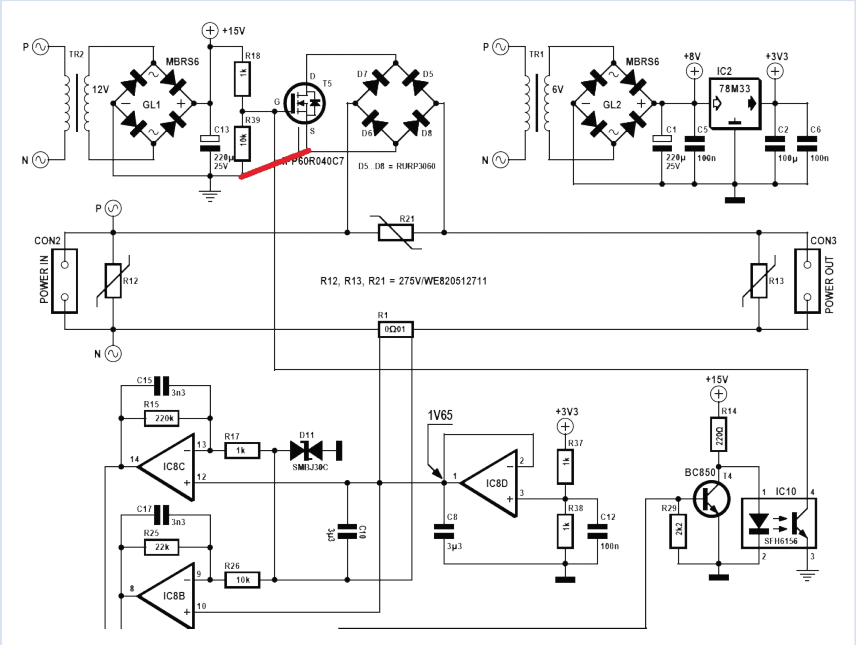
Ralf Schmiedel et **Jens Nickel** (Elektor)

Zoom sur les transformateurs secteur

Elektor 09-10/2021, p. 34

(200587-04)

Une connexion à la terre est absente du schéma du banc d'essai présenté à la page 35 (voir figure ci-contre).

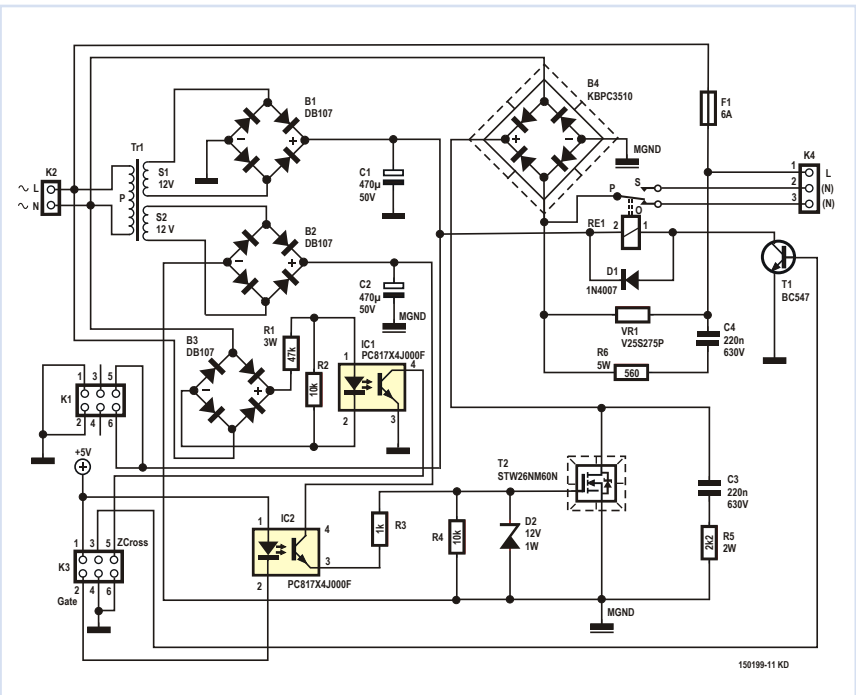


Variateur pour moteur monophasé 1 kW

Elektor 01-02/2020, p. 70

(191142-04)

Il y a une erreur dans le schéma de la carte de puissance du variateur pour moteur monophasé de 1 kW présenté à la page 72 du numéro de janvier/février 2020. Les diodes du pont redresseur B4 sont dessinées dans le mauvais sens. Pour que le dessin soit correct, le symbole des quatre diodes doit être tourné à 180°. Les symboles « + » et « – » du schéma sont corrects et le circuit imprimé l'est également.

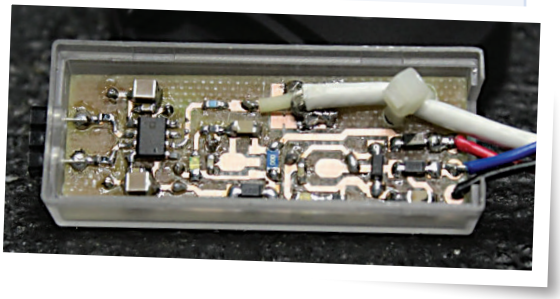


Sonde de courant différentielle pour oscillo – 2.0

Elektor 11-12/2020, p. 68 (180665-04)

L'INA849 de TI est une meilleure alternative, compatible broche à broche, que l'AD8421 utilisé dans cette sonde. Il ne nécessite aucune compensation de la réponse en fréquence et possède une bande passante nettement supérieure. Il est seulement nécessaire d'ajuster la valeur de la résistance qui définit le gain.

Alfred Rosenkränzer



Testeur multifonction LCR-T7 de Joy-IT

Elektor 11-12/2021, p. 22 (210365-04)

L'article donne l'impression que Joy-IT a développé lui-même l'appareil décrit. Ce n'est absolument pas le cas. Le LCR-T7 est une copie d'un projet communautaire basé sur un projet *open source* (www.mikrocontroller.net/articles/AVR_Transistortester). L'idée d'un testeur automatique de transistors est venue à l'origine de Markus Frejek, elle a été reprise par plusieurs autres passionnés d'électronique tels que Markus Reschke, Karl-Heinz Kübbeler, Pieter-Tjerk de Boer et bien d'autres qui ont amélioré le montage en y apportant leur expertise et en déployant beaucoup d'efforts. Il existe également un forum réservé aux testeurs (www.mikrocontroller.net/topic/transistortester-avr), où l'évolution de l'appareil est documentée depuis 2012. Aucune référence à cette ressource n'est faite dans l'article d'*Elektor*. Je suppose que cette omission n'est pas délibérée, mais résulte d'une connaissance incomplète de votre part. La propriété intellectuelle, le plagiat et le droit d'auteur font actuellement l'objet de nombreuses discussions. Il n'y a aucune indication qu'une loi ait été violée, pour autant que je sache, les auteurs originaux de cet instrument copié n'ont pas revendiqué de droits d'auteur ou cherché une compensation financière pour l'exploitation commerciale de leurs idées. L'équité veut que l'on rende hommage à ceux qui le méritent.

Marcel « Derri » Derrmann

Cher M. Derrmann,
merci pour vos informations détaillées. Vous avez bien sûr raison de dire que nous n'avons pas cherché à réinventer la roue. Les développeurs originaux de ce projet méritent un grand respect pour leur idée fantastique et le travail qu'ils ont fourni pour sa réalisation. Notre société ne se contente pas de développer ses propres appareils, mais elle reprend également de grandes idées et essaie de les transformer en un produit fini de qualité pour les développeurs et les makers, au prix le plus attractif possible. Notre département de développement est souvent responsable de l'ajout de la « touche finale ».

Volker Bode, SiMAC/Joy-IT

Centrale solaire sur balcon

Elektor 09-10/2021, p. 44 (210326-04)

Vous voulez utiliser les prises SmartHome pour mesurer l'alimentation en énergie solaire ? Notre lecteur Jörg Trautmann a voulu le faire et partage son expérience :

Sur mon installation photovoltaïque, j'ai été un peu déçu parce que les premières prises SmartHome que j'ai testées ne pouvaient mesurer que la puissance consommée par un appareil et n'indiquaient pas la puissance injectée dans la prise par un onduleur et un système photovoltaïque. Après une longue période de recherche, j'ai découvert la prise SmartHome LogiSmart PA0066 de LogiLink, qui peut réellement mesurer la puissance injectée dans le réseau. Pour autant que je sache, la méthode la plus simple pour mesurer le courant est d'utiliser une pince ampèremétrique ; le sens du courant n'a aucune importance pour la mesure. Ici, les appareils devraient utiliser une méthode de mesure différente, ce qui constitue un aspect très intéressant de leur conception.

Jörg Trautmann

N.B. La prise intelligente à laquelle il est fait référence accepte les prises de courant domestiques françaises standard. Au Royaume-Uni, un onduleur de réseau ne doit être connecté au câblage du réseau que via son propre isolateur.

Des questions, des commentaires ?
Contactez Elektor (redaction@elektor.fr).

210560-04

de la couleur au son

Comment exploiter un capteur de couleurs avec l'I²C

Annika Schlechter (Allemagne) et
Volker Ziemann (Suède)

L'électronique peut être utile pour les personnes handicapées, comme le montre ce projet d'étudiants. Pour les personnes qui ne peuvent pas voir les couleurs, un capteur de couleurs peut assurer cette fonction : un vibreur est utilisé comme retour acoustique pour les informations de couleur et de luminosité. Dans cet article, nous présentons deux versions du projet basé sur un Arduino Nano, une carte bon marché.

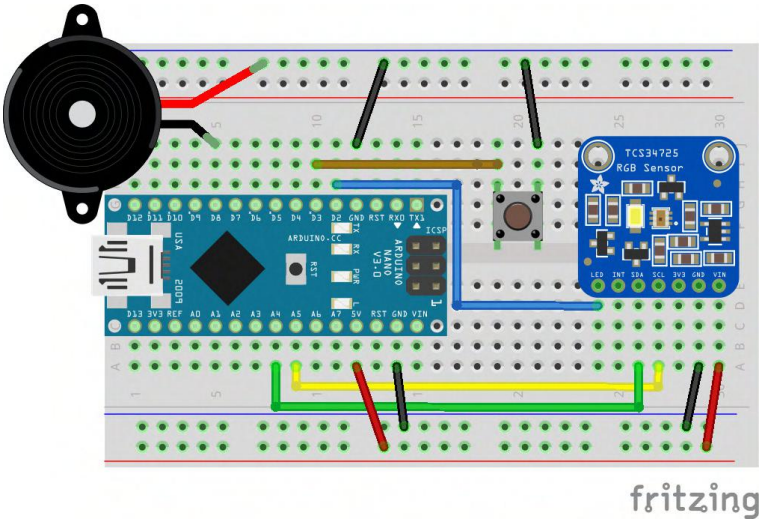


Figure 1 : câblage du prototype. Un vibreur piézoélectrique et un capteur de couleurs TCS34725 monté sur une petite carte de liaison sont connectés à un Arduino Nano.

Comment une personne malvoyante peut-elle choisir le matin la couleur de ses vêtements pour une tenue assortie ? Et bien, en convertissant l'information sur la couleur en un signal sonore. C'est l'idée de ce projet. La couleur est convertie en hauteur du son et la luminosité en durée du son.

La figure 1 montre le câblage du prototype avec un vibreur piézoélectrique et un capteur de couleurs ams TCS34725 monté sur une petite carte de liaison [1]. Les deux composants sont connectés à un Arduino Nano. Ce prototype a été mis au point dans le cadre du cours « From Sensor to Report » à l'Université d'Uppsala (Suède) [2] qui enseigne au grand public les compétences de base en acquisition de données, notamment l'interfaçage des capteurs et des microcontrôleurs. C'est pourquoi nous n'utilisons pas de bibliothèques prêtes à l'emploi pour l'interface avec le capteur, mais nous codons nous-mêmes les fonctions de base pour extraire les données des capteurs. De plus, la logique d'assignation des sons aux couleurs réside initialement dans un script Python sur un ordinateur portable, et non pas dans le microcontrôleur. Le PC et le microcontrôleur communiquent par l'intermédiaire d'une interface série. Nous avons mis au point un protocole de communication pour envoyer les valeurs du Nano vers le PC, tandis que les commandes vont en sens inverse pour activer le vibreur avec la bonne fréquence. Ensuite, nous avons amélioré le prototype pour permettre un fonctionnement autonome sur le Nano. Néanmoins, il ne s'agit pas d'un produit prêt pour une mise en production, mais d'un prototype qui met en œuvre la fonction de base et invite le lecteur à poursuivre l'expérimentation.

Avant de commencer à assembler les composants, il faut installer l'environnement de développement (EDI) Arduino [3] pour créer les programmes du Nano. L'EDI est disponible pour tous les systèmes d'exploitation fréquemment utilisés. Suivez les instructions correspondant à votre système. Si vous n'avez jamais utilisé de cartes Arduino auparavant, consultez les tutoriels Arduino [4] afin de vous familiariser avec l'EDI Arduino. Comme second prérequis, vous devez installer Python [5], également disponible pour tous les systèmes. Les fichiers binaires Windows et MAC OS sont disponibles sur le site web de Python [5] et pratiquement toutes les distributions Linux en ont des *packages* dans leurs dépôts. À nouveau, suivez les instructions pour votre système. Mais, assez de préparatifs, il est temps de commencer !

Le capteur de couleurs

Ce projet repose sur le capteur de couleurs TCS34725 [6] qui comprend un réseau de 3×4 photodiodes (**figure 2**). Les filtres photosensibles installés devant les diodes les rendent sensibles à la lumière rouge, verte et bleue. De plus, une diode claire dépourvue de filtre fournit des

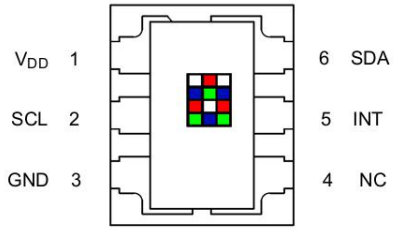


Figure 2 : le capteur de couleurs TCS34725 intègre un réseau de 3×4 photodiodes. Les filtres photosensibles installés devant les diodes rendent celles-ci sensibles à la lumière rouge, verte et bleue. Source : fiche technique / ams [6].

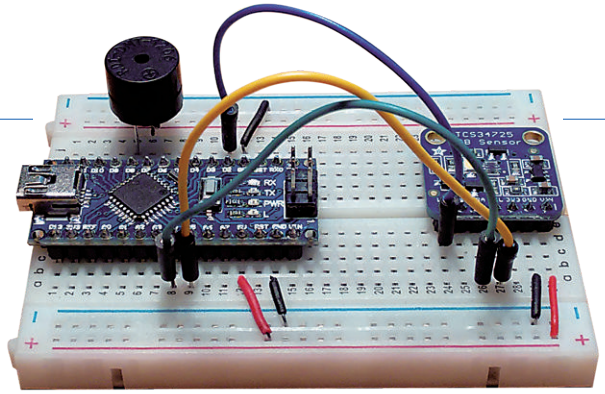


Figure 3 : le matériel (voir figure 1) est assemblé sur une platine d'expérimentation (ici sans le bouton).

PRODUITS

- **Arduino Nano**
  www.elektor.fr/arduino-nano
- **Platine d'expérimentation (830 points)**
  www.elektor.fr/breadboard-830-tie-points

informations sur la luminosité. Le signal analogique issu des diodes est converti sous forme numérique grâce aux convertisseurs analogique-numérique embarqués. L'information sur la couleur est disponible grâce à une interface I²C reliée à un microcontrôleur à l'aide uniquement de deux fils pour la communication : un pour le signal d'horloge (SLC) et un autre pour les données (SDA). Le capteur de couleurs est ensuite commandé par l'écriture et la lecture de ses registres. Mais laissons cela pour plus tard, lorsque nous décrirons le logiciel.

Le matériel est assemblé sur une platine d'expérimentation (**figure 3**), de manière assez similaire à la figure 1 avec le Nano visible du côté gauche et le capteur de couleurs TCS34725 sur la droite. Ce dernier est relié avec des fils rouges et noirs à la masse et au rail de 5 V en bas. De même, les broches correspondantes du Nano sont reliées au rail d'alimentation. Les fils verts et jaunes relient les broches de données (SDA) et d'horloge (SCL) aux broches correspondantes sur le Nano, situées en A4 et A5 respectivement. Le fil bleu qui relie la broche de sortie numérique D2 à la broche étiquetée *LED* sur le capteur de couleurs permet de commander une LED blanche montée sur la carte de liaison. Le vibreur est connecté à la broche D8 sur le Nano. Le bouton qui relie la broche D3 à la masse avec les fils marron déclenche une conversion lors le Nano fonctionne en mode autonome, c'est expliqué ci-dessous.

Ce circuit s'anime grâce à un croquis Arduino qui inclue en premier lieu la bibliothèque *wire.h* pour fournir des fonctions de bas niveau afin de communiquer dans les lignes de l'I²C. Ces fonctions sont utilisées immédiatement dans nos deux fonctions `I2Cread()` et `I2Cwrite()`, qui englobent toutes les communications avec le capteur.

La fonction `I2Cread()` sert à extraire une valeur ou un paramètre du capteur, stockés dans ce que l'on appelle des registres. Comme paramètres, la fonction reçoit l'adresse du capteur sur le bus I²C, ici `0x29` et l'adresse du registre (il nous a fallu du temps pour comprendre qu'il faut toujours lui ajouter 0x80). La fonction retourne le contenu de ce registre.

La fonction `I2Cwrite()` est utilisée pour envoyer des commandes du contrôleur vers le capteur, par exemple pour régler les valeurs de configuration. Cette fonction reçoit l'adresse du capteur et du registre ainsi que la nouvelle valeur qui doit être inscrite dans ce registre. Cette fonction ne retourne aucune valeur (`void`).

L'application

Revenons à notre croquis spécifique. Dans la fonction `setup()`, nous initialisons la communication série, les fonctions I²C de base et le capteur (par l'intermédiaire de la fonction `color_begin()`). Dans la boucle principale `loop()`, exécutée indéfiniment, nous vérifions en premier lieu si une commande est arrivée du PC sur la ligne série. Si c'est le cas, le contenu des registres du capteur est lu pour les trois couleurs. D'après la fiche technique, pour chaque couleur, il faut lire deux registres à 8 bits et assembler les octets reçus en un mot de 16 bits. Voici ce que cela donne la couleur rouge :

```
b3=I2Cread(TCS34725,0x16);   // raw data, red
b4=I2Cread(TCS34725,0x17);
red=b4*256+b3;
```

Ici `b3` contient l'octet de poids faible et `b4` l'octet de poids fort, qui est multiplié par 256 et ajouté à `b3` pour obtenir la variable à 16 bits `red`. Nous traitons les autres couleurs de la même façon ; seules les adresses des registres sont adaptées.

De plus, nous lisons la luminosité mesurée par la diode non filtrée et la stockons dans la variable `clea`. Cela déterminera la durée `d` du son. Si la valeur de `clea` est inférieure à 1000, le son durera 0,5 s ; si elle est plus élevée, ce sera 1,5 s. Ensuite la commande reçue est lue et le résultat est copié dans le tableau de caractères `line`, on peut ainsi utiliser la fonction standard du langage C `strstr()` afin de déterminer quelle commande a été envoyée. Ceci est illustré dans l'extrait de code suivant :

```
Serial.readStringUntil('\n').toCharArray(line,30);
if (strstr(line,"color?")==line) {
  Serial.print(clea); Serial.print(" ");
  Serial.print(red); Serial.print(" ");
  Serial.print(green); Serial.print(" ");
  Serial.println(blue);
} else if (strstr(line,"tone_red")==line) {
  tone(buzz, 262, d);
...
```

On peut voir que la commande *color?* entraîne le renvoi par le Nano sur la ligne série des valeurs des quatre signaux : `clea`, `red`, `green` et `blue`. Si *tone_red* est reçu, la commande intégrée `tone()` est utilisée pour activer le vibreur piézoélectrique, dans le cas présent avec la fréquence qui est assignée à la couleur rouge (*red*). Plusieurs blocs `strstr()` construits de la même façon suivent et entraînent les actions appropriées.

Notez que nous utilisons un protocole simple, basé sur des allers-retours de chaînes de caractères ; une demande de l'ordinateur portable se termine par un point d'interrogation, par exemple « *color?* » et le Nano répond en renvoyant les valeurs. Une instruction est basée sur le simple envoi de la commande, par exemple « *tone_red* » qui provoque l'émission d'un son par le vibreur à 262 Hz pendant la durée spécifiée par `d`. La communication est à peu près équivalente au langage de commande SCPI qui est pris en charge par les oscilloscopes modernes et d'autres appareils de test et de mesure. Cela facilite l'interfaçage avec des programmes externes qui prennent en charge l'accès à la ligne série ; c'est le cas de Python, Octave, Matlab et Labview.

Le programme sur PC

Nous utilisons Python 3 sur l'ordinateur portable pour communiquer avec le Nano. Le **listing 1** présente le code assez élémentaire.

Après avoir importé les bibliothèques nécessaires pour la communication série et la synchronisation, le port série sur lequel le Nano est connecté est ouvert à une vitesse de transmission (baud) correspondant à celle choisie sur le Nano. Trois secondes sont allouées au système d'exploitation pour établir la connexion, puis la commande « *color?* » est envoyée.
Notez que Python 3 code les chaînes de caractères en Unicode, tandis que le Nano attend de simples chaînes ASCII. Il faut par conséquent ajouter la lettre « b » à la commande « *color?* » afin d'envoyer la chaîne en tant que simples données ASCII. Après un autre court temps d'attente, la réponse envoyée par le Nano est stockée dans la variable `reply` qui contient les valeurs des quatre couleurs. La méthode `.split()` permet d'extraire chaque valeur et de lui affecter un nom mnémonique, tel que `red`. Comme nous disposons des valeurs, nous sommes prêts à mettre en œuvre la logique d'assignation des sons aux couleurs. Cette assignation est basée sur une expérimentation assez large avec des feuilles de papier colorées jusqu'à l'obtention d'un réglage correct. Nous avons utilisé des papiers avec une teinte claire, une teinte neutre et une teinte foncée pour chacune des trois couleurs de base pour étalonner le capteur. Nous avons mesuré à plusieurs reprises les valeurs de couleur respectives à une distance fixe et nous avons utilisé ces informations pour préciser les constantes dans le code Python. Par exemple, la commande *tone_red* est envoyée au Nano si la composante rouge, normalisée par rapport à l'intensité, est supérieure à 0,6. Ensuite l'intensité du vert est testée et la commande *tone_green* est envoyée si le seuil est dépassé. Enfin, la valeur de la composante bleue est testée. Si aucun seuil n'est atteint, « *no_color* » est envoyé au Nano, le faisant émettre la tonalité correspondante. Vous devrez probablement effectuer quelques expérimentations pour déterminer les constantes et adapter le système à votre garde-robe.

Version autonome

La version autonome du code est disponible sur la page web de cet article [7]. Elle respecte assez fidèlement l'exemple ci-dessus, sauf qu'au lieu d'attendre l'arrivée d'une commande sur la ligne série, le Nano attend à présent que l'on presse sur le bouton avant de lire le capteur et d'émettre le son adapté. De plus, l'information sur les couleurs est directement traitée dans le Nano, comme le montre l'extrait de code suivant :

```
if (red/clea > 0.6) {
  tone(buzz, 262, d);
} else if (green/clea > 0.3) {
...
```

Comme il est préférable d'éclairer systématiquement les vêtements avec la LED intégrée à la carte de liaison, il est possible de l'allumer avec `digitalWrite(led,HIGH)` juste avant la lecture des registres du TCS34725 et de l'éteindre juste après.
Enfin, il est possible de rendre le système portatif en utilisant un simple ATmega328, flashé avec un chargeur d'amorçage Arduino et relié au capteur. Alimenté par une batterie tant que l'on presse sur le bouton, le système se met en route et convertit continuellement la couleur en son jusqu'à ce que le bouton soit relâché afin de déconnecter la batterie. Le logiciel est également disponible sur la page web de l'article. Nous laissons la

À propos des auteurs

L'intérêt de Volker Ziemann pour l'électronique prend ses racines dans *Elektor* à l'ère de l'amplificateur Edwin 40W (milieu des années 70), mais il a en parallèle depuis cette époque étudié la physique et travaillé dans le domaine des accélérateurs de particules (SLAC aux États-Unis, CERN à Genève et maintenant à Uppsala en Suède). Étant donné que l'électronique joue un rôle important dans le contrôle et l'acquisition de données, son centre d'intérêt de toujours lui a été très utile au cours de sa carrière. Il enseigne à présent à l'université d'Uppsala et il est l'auteur de trois livres, l'un portant sur l'acquisition de données avec les cartes Arduino et le Raspberry Pi.
Annika Schlechter étudie afin d'obtenir un master en physique à l'université d'Heidelberg en Allemagne. Elle effectue actuellement un stage à l'Institut de recherche sur le cancer d'Heidelberg dans le cadre de sa licence. Le projet est né lors d'un échange Erasmus à l'université d'Uppsala en Suède au cours duquel Annika a participé au cours de Volker intitulé « Sensor to Report ».

Listing 1 : script Python pour l'ordinateur portable communiquant avec l'Arduino Nano.

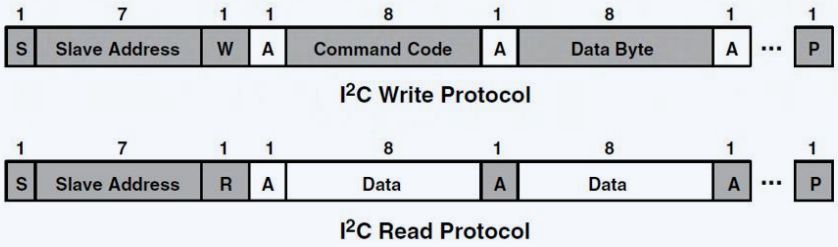
```
# colortosound_arduino.py
import serial, time

ser=serial.Serial("/dev/ttyUSB0",9600,timeout=1)
time.sleep(3);

ser.write(b"color?\n")  # write command to get colors
time.sleep(2);

reply=ser.readlines()   # read answer

# process answer to get color values
colors = reply[0].split()
for i, entry in enumerate(colors):
    colors[i]= int(entry.decode('utf-8').replace('\r\n', ''))

intensity = colors [0]; red = colors [1]
green = colors [2]; blue = colors [3]

# write command to control buzzer according to color

if red/intensity > 0.6:
    ser.write(b"tone_red\n")
elif green/intensity > 0.3:
    ser.write(b"tone_green\n")
elif blue/intensity > 0.24:
    ser.write(b"tone_blue\n")
else:
    ser.write(b"no_color\n")

ser.close()
```

Communication I²C

La communication entre deux circuits intégrés (I²C) est basée sur un protocole série synchrone et requiert deux fils. La communication suit une règle simple : détecter le niveau de tension dans l'un des fils, appelé SDA, après que la tension sur l'autre fil, appelé SCL, a été modifiée. Une lecture de ce type fournit un seul bit, huit bits consécutifs constituant un octet. Un des éléments du bus I²C, habituellement un microcontrôleur, orchestre la communication et fournit toujours le signal d'horloge. Les autres éléments ont la possibilité de participer à la communication si on s'adresse à eux et qu'on leur donne l'autorisation d'envoyer leurs données sur la ligne SDA. De plus, après chaque octet, on laisse au capteur un cycle d'horloge pour confirmer la réception de l'octet précédent.

Pour lancer la communication avec un capteur, le microcontrôleur envoie un octet. Les sept premiers bits de cet octet contiennent l'adresse tandis que le dernier bit indique si le contrôleur ou le capteur a le contrôle de la ligne SDA pour transmettre l'octet suivant. Si le contrôleur souhaite continuer l'envoi, par exemple pour modifier un registre dans le capteur, il envoie deux octets supplémentaires, l'adresse du registre et la nouvelle valeur. Si, par contre, le contrôleur veut lire un registre du capteur, il transmet le bit pour indiquer que le capteur a la main sur la ligne SDA ; le contrôleur continue alors à délivrer le signal d'horloge et lit la ligne SDA au rythme du signal d'horloge.

L'image du haut de la figure illustre l'écriture dans le capteur. Les bits qui sont grisés sont contrôlés par le microcontrôleur tandis que les bits qui ne sont pas grisés le sont par le capteur. Pour lancer une transaction, le microcontrôleur crée une condition de démarrage, désignée par « S », suivie des sept bits de l'adresse du capteur et du huitième bit activé pour indiquer que le contrôleur veut poursuivre l'envoi. Un bit « A » est retourné par le capteur pour indiquer qu'il a compris. Le second octet contient l'adresse du registre sur le capteur et un octet avec la valeur à inscrire dans le registre. Remarquez bien le bit noté « A » avec lequel le capteur confirme la réception de chaque octet. La lecture du capteur, illustrée dans l'image du bas, fonctionne à peu près de la même façon, simplement les huit bits du premier octet indiquent à présent que le capteur est autorisé à « parler » et que la ligne SDA est alors contrôlée par le capteur, tandis que le microcontrôleur doit confirmer chaque octet reçu.



Source : fiche technique TCS34725 / ams [6].

construction de cette version à nos lecteurs enthousiastes.
Pendant le cours « Sensor to Report », le capteur de couleurs a suscité l'intérêt. Une étudiante qui souhaitait surveiller la couleur du ciel sur une période prolongée afin d'analyser ensuite les données, a utilisé pratiquement la même configuration avec le Nano communiquant avec l'ordinateur hôte par l'intermédiaire de la ligne série. Elle a utilisé un script en Python sur un Raspberry Pi pour stocker les données dans une base de données MySQL, également hébergée sur le Raspberry Pi. Elle a ensuite utilisé Octave pour préparer les courbes des couleurs enregistrées pendant environ une semaine. Les courbes ont été enregistrées juste avant Noël et cela nous a tristement rappelé à quel point les jours peuvent être courts à Uppsala. D'autres projets d'acquisition de données dans le même esprit sont expliqués sur la page [8].

210051-04

Contributeurs

Conception et texte : **Annika Schlechter**, **Volker Ziemann**
Rédaction : **Jens Nickel**
Mise en page : **Sylvia Sopamena**

Des questions, des commentaires ?

Contactez Elektor (redaction@elektor.fr).

LIENS

[1] Carte de liaison TCS34725 : https://www.adafruit.com/product/1334
[2] Cours « Sensor to Report » : https://ziemann.web.cern.ch/ziemann/teaching/s2r19/
[3] Site web Arduino : https://www.arduino.cc/
[4] Tutoriels Arduino : https://www.arduino.cc/en/Tutorial/HomePage
[5] Site web Python : https://www.python.org/
[6] Fiche technique du capteur de couleurs TCS34725 : https://ams.com/tcs34725#tab/documents
[7] Téléchargement du logiciel : http://www.elektormagazine.fr/210051-04
[8] V. Ziemann, « A Hands-on Course on Sensors using the Arduino and Raspberry Pi », CRC Press, 2018.

sourcemètre **BattLab-One**

Mesurer et optimiser la durée de vie des batteries des appareils IdO

Tam Hanna (Slovaquie)

Pour choisir une batterie appropriée incorporée dans un appareil autonome relié à l'Internet des objets (IdO), vous pouvez utiliser un multimètre numérique pour mesurer le courant actif et le courant de veille. Vous pourrez ainsi estimer l'autonomie de l'appareil avec une charge, optimiser la conception en modifiant le rapport des modes veille/activité, voire même opter pour une capacité ou un type de batteries différents. Raccordé via un port USB, le BattLab One peut faire tout cela d'un simple clic de souris. Il alimente l'appareil testé et affiche toutes les données relatives à l'autonomie prévue de la batterie…

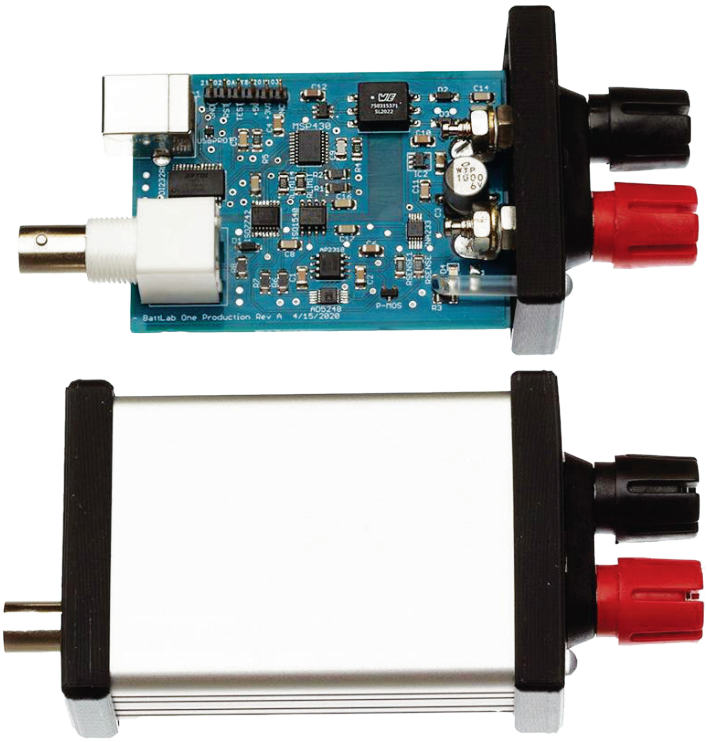


Figure 1. Le BattLab-One et son boîtier en aluminium.

Obtenir une mesure fiable de l'énergie consommée par un appareil doté d'un microcontrôleur n'est pas anodin. Comme pour toute puce informatique, la consommation d'énergie dépend surtout de la charge de traitement instantanée et de l'état opérationnel. Il ne suffit pas de prendre en compte le fonctionnement normal et les états de veille ou de sommeil profond, dans lesquels la consommation de courant des microcontrôleurs les plus diffusés peut varier de moins de 1 µA à plusieurs centaines de milliampères (mA). Certains périphériques du dispositif peuvent également n'être actifs qu'à certains moments, de sorte que la consommation de courant ne sera pas constante. Un outil spécialisé comme BattLab-One est très utile à cet égard et peut tenir compte de ce comportement dynamique pour faire une estimation raisonnablement réaliste de la durée de fonctionnement de l'appareil lorsqu'il est alimenté par différents types de batteries.

Qu'est-ce que BattLab-One ?

Dans le domaine de la mesure de la consommation électrique, un sourcemètre (SMU) est un outil très utile dans votre labo. Ce type d'alimentation apporte du courant au dispositif testé et effectue des mesures précises de la puissance consommée sur des périodes définies. Parmi des exemples de sourcemètres professionnels figurent les appareils fabriqués par Keithley. Le BattLab-One de Bluebird Labs remplit une fonction similaire, mais pour un prix inférieur à 100 €. L'instrument (y compris le boîtier métallique) est disponible dans la boutique Elektor [1].

Reportez-vous à la **figure 1** pour voir la prise USB de type B sur la gauche ainsi qu'une prise BNC acceptant un signal d'entrée de déclenchement. Sur la droite se trouvent les deux prises de 4 mm servant à alimenter l'appareil testé. La **figure 2** montre les garnitures en plastique à dévisser pour fixer les pinces crocodiles aux bornes. Un examen plus attentif des coiffes d'extrémité du boîtier montre une texture de surface adaptée à l'impression 3D.

BattLab-One est alimenté par le port USB d'un ordinateur et peut délivrer une gamme de tensions d'alimentation comprises entre 1,2 à 4,5 V et des courants atteignant 450 mA maximum. Les profils de décharge des types de batteries fréquemment utilisés (comme les technologies Li-Ion, $LiFePO_4$ alcaline, NiMh, NiCd) sont inclus dans le logiciel et utilisés pour calculer les prévisions de durée de vie. BattLab-One ne tient pas compte de certaines subtilités comme

les chutes de tension liées à la charge et l'âge de la batterie. Selon le fabricant, l'appareil de mesure offre une simulation de base de la batterie et une plage de mesure de courant comprise entre 10 µA et 500 mA. Un CA/N de 16 bits avec échantillonnage à 1 kHz est incorporé pour mesurer le courant lié au dispositif testé.

BattLab-One et Linux

Le logiciel de l'instrument a été développé par Bluebird Labs en Python. Pour ma première tentative, j'ai voulu faire fonctionner le système directement sous Ubuntu 20.04 LTS. Le code source du logiciel est disponible sur GitHub [2] et peut être compilé à l'aide des méthodes habituelles. Cependant, Bluebird Labs a adopté une approche différente en utilisant *pipenv*. Tout utilisateur qui, comme moi, ne dispose que de la variante habituelle de *venv* doit installer *pipenv*. Pour moi, le processus s'est déroulé comme suit :

Figure 2. Branchement électrique des pinces crocodiles.

```
tamhan@TAMHAN18:~/BattLabonespace/
BattLab-One$ sudo apt install pipenv
```

Les premiers problèmes sont apparus lors de la configuration de l'environnement virtuel, car elle suppose la version 3.7. de Python. Si, comme moi, vous voulez utiliser une version plus récente (c'est-à-dire la version 3.8 préinstallée sous Ubuntu 20.04 LTS), il faudra ajouter la déclaration suivante :

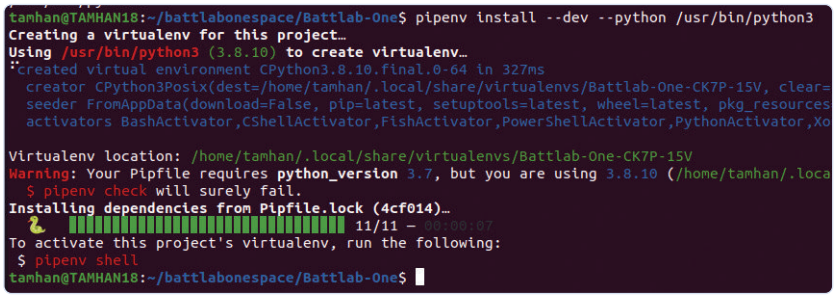


Figure 3. Message d'erreur PipEnv.

```
tamhan@TAMHAN18:~/BattLabonespace/
BattLab-One$ which python3
/usr/bin/python3
tamhan@TAMHAN18:~/BattLabonespace/BattLab-One$ pipenv
install --dev --python /usr/bin/python3
```

Considérons maintenant le processus de création d'un *virtualenv* pour ce projet. Le logiciel émet des avertissements (**fig. 3**) lors de la saisie des commandes. L'environnement virtuel est donc activé selon le schéma suivant. Il restera actif comme un environnement *venv* normal jusqu'à ce que la fenêtre du terminal soit fermée :

```
tamhan@TAMHAN18:~/BattLabonespace/BattLab-One$ pipenv
shell
. . .
```

L'exécution des trois fichiers *.py* avec le logiciel est maintenant possible en utilisant la méthode suivante :

```
(BattLab-One-CK7P-15V) tamhan@TAMHAN18:~/
BattLabonespace/BattLab-One$ python3
BattLab-Release_V1.2.1.py
Traceback (most recent call last):
  File "BattLab-Release_V1.2.1.py", line 59, in
<module>
    import pkg_resources.py2_warn
ModuleNotFoundError: No module named 'pkg_resources.
py2_warn'
```

Malheureusement, Bluebird Labs ne se conforme pas à la spécification des outils d'installation, mais inclut explicitement une bibliothèque qui n'est disponible que dans certaines versions de ces outils. Le fichier *.py* doit être modifié pour supprimer l'instruction suivante :

```
import pkg_resources.py2_warn
```

Le logiciel a aussi un problème concernant les chemins d'accès locaux :

```
(BattLab-One-CK7P-15V) tamhan@TAMHAN18:~/
BattLabonespace/BattLab-One$ python
BattLab-Release_V1.2.1.py
. . .
_tkinter.TclError: error reading bitmap file "icons\
bbirdlogo.xbm"
```

La solution consiste à ouvrir l'interface graphique de TK-Inter en supprimant la ligne :

```
root.iconbitmap(bitmap=GetIconPath())
```

pour supprimer l'icône du programme. Par la même occasion, vous pouvez adapter le chemin d'accès au fichier de la manière suivante :

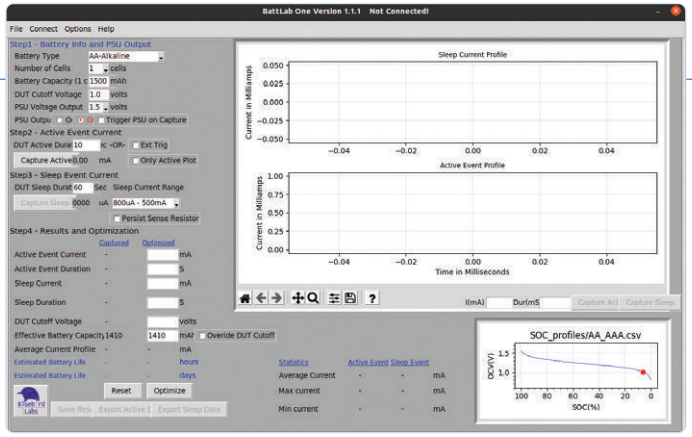


Figure 4. L'interface graphique Linux BattLab.

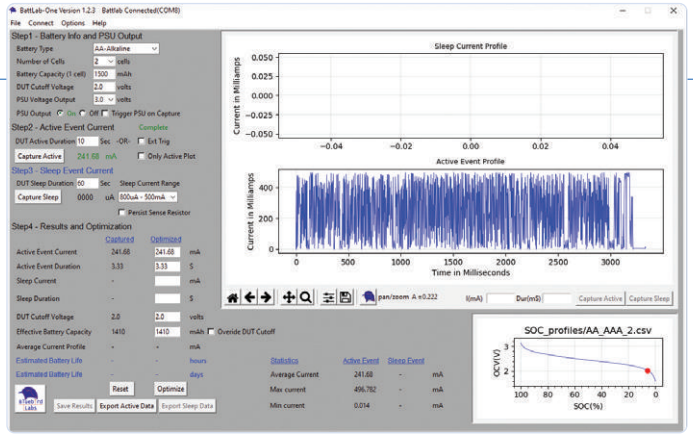


Figure 5. BattLab-One enregistre et affiche la consommation de courant du dispositif testé.

```
img = PhotoImage(file='/home/tamhan/BattLabonespace/
BattLab-One/icons/bbirdlogo_png1.png')
```

La tentative suivante d'exécution montre que l'environnement TK-Inter recherche le fichier *bblogo.gif* dans le mauvais dossier.

```
_tkinter.TclError: couldn't open "/home/tamhan/.
local/share/virtualenvs/BattLab-One-CK7P-15V/lib/
python3.8/site-packages/matplotlib/mpl-data/images/
bblogo.gif": no such file or directory
```

Vous pouvez donc copier n'importe quel fichier GIF dans ce répertoire et lui donner le nom approprié. Si vous mettez la ligne suivante en commentaire, vous pouvez démarrer le programme avec une gamme de fonctions légèrement limitée (**fig. 4**) :

```
#toolbar.children['!button8'].
config(command=select_range)
```

La désactivation du bouton de la plage de mesure s'avère peu critique dans la pratique, car le diagramme MatPlotLib est de toute façon problématique en termes de convivialité. De nombreuses étiquettes apparaissent tronquées et la taille de la fenêtre ne peut être ajustée. Comme nous allons le voir, le programme a moins de problèmes à fonctionner sous Windows 10. Avant de changer de système d'exploitation, nous pouvons faire le ménage pour supprimer l'environnement virtuel créé et ainsi libérer de l'espace de stockage de masse :

```
(BattLab-One-CK7P-15V) tamhan@TAMHAN18:~/
BattLabonespace/BattLab-One$ exit
exit
tamhan@TAMHAN18:~/BattLabonespace/BattLab-
One$ cd /home/tamhan/.local/share/virtualenvs/
BattLab-One-CK7P-15V
tamhan@TAMHAN18:~/.local/share/virtualenvs/BattLab-
One-CK7P-15V$ rm * -rf
tamhan@TAMHAN18:~/.local/share/virtualenvs/BattLab-
One-CK7P-15V$ ls
```

BattLab-One fonctionne sous Windows 10

La situation est plus simple dans la mesure où des fichiers *.exe* prêts à l'emploi sont fournis sous [3]. L'archive ZIP comprend, entre autres, le sous-dossier *SOC_profiles*, qui contient les différents profils de batterie sous forme de tableau. Vous devez dézipper l'archive.

Lorsque le fichier *BattLab-Release_V1.2.3.exe* est exécuté, l'affichage de l'interface utilisateur se présente sous la forme de la version Linux de la **figure 4**. La version Windows se comporte en grande partie de la même manière, mais présente moins d'erreurs de rendu. Tout d'abord, nous pouvons sélectionner le type de batterie dans la zone supérieure gauche. Les données de batterie associées servent à créer la courbe de décharge caractéristique présentée en bas à droite. À droite du champ *PSU Output*, les boutons *On* et *Off* permettent de commuter le convertisseur de tension intégré pour alimenter l'appareil testé connecté aux deux prises de sortie.

Je travaille depuis quelque temps sur mon propre projet de capteur IdO appelé *HygroSage*. Avec son écran couleur et son processeur puissant, il serait un candidat idéal pour tester le BattLab-One. J'ai connecté le capteur. En cliquant sur l'option *Capture active*, le logiciel affiche une barre de progression verte clignotante intitulée *Active Event Current*, puis ne fait rien de visible pendant 10 s. Pendant ce temps, le système mesure les informations et finit par les afficher comme le montre la **figure 5**.

Le curseur permet d'afficher la valeur de n'importe quel point de la forme d'onde. En pratique, le fonctionnement n'est pas aussi convaincant car la valeur indiquée dans la zone de texte ci-dessous ne montre pas la valeur mesurée, mais la position du pointeur de la souris sur le graphique. Vous devez donc déplacer le pointeur de la souris vers un point de la courbe. En outre, il n'y a pas de moyen vraiment pratique de zoomer dans le diagramme (produit par MatPlotLib). Malgré la résolution « full HD » de mon écran, je n'ai pas pu agrandir la fenêtre de l'interface utilisateur. Il est cependant possible de sélectionner une zone de la forme d'onde en utilisant l'option *Zoom to Rectangle*.

Pour effectuer un test, j'ai connecté un multimètre étalonné de haute qualité en série avec l'appareil testé. J'ai ainsi pu mesurer un courant d'environ 32 mA. Mais BattLab-One ne partageait pas cet avis. Il semble que le régulateur à découpage de type MCP1640 du BattLab-One produit un bruit qui interfère réellement avec son fonctionnement. Comme deuxième test, j'ai utilisé une résistance de 1 kΩ comme charge et spécifié une tension de sortie de 3 V. Le résultat est la trace représentant le bruit de quantification des mesures dans la

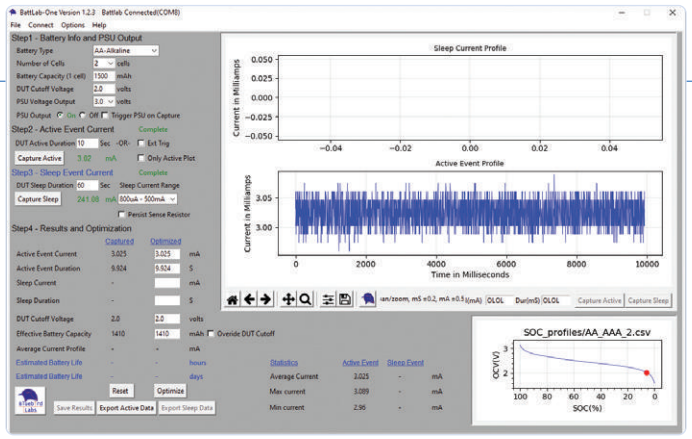


Figure 6. Trace montrant le courant circulant dans une résistance de 1 kΩ sous 3 V avec un certain niveau de bruit apparent.

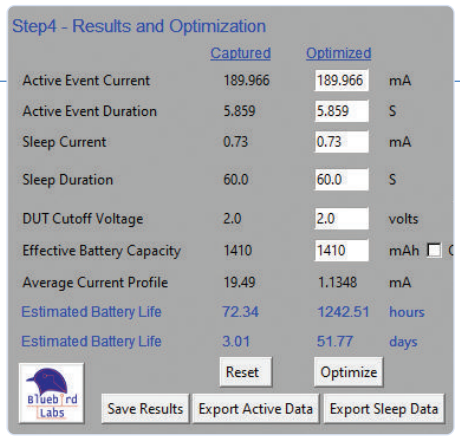


Figure 7. L'autonomie de la batterie peut être optimisée en utilisant les valeurs de courant en mode veille et en mode actif et la capacité de la batterie.

figure 6. La connexion d'un condensateur électrolytique de 100 µF en parallèle n'a pas entraîné d'amélioration significative de la forme d'onde de consommation de courant quantifiée.

Courant de veille et optimisation

Une fois la consommation de courant actif obtenue à l'étape 2, nous pouvons passer à l'étape 3 pour mesurer le courant de veille. Ici, nous devons nous assurer que le dispositif testé sera en mode veille au moment où la mesure sera effectuée. Nous allons entrer les paramètres comme nous l'avons fait à l'étape 2 pour la capture du mode actif. Sauf si le projet conçu ne fonctionne pas correctement, l'appareil consommera toujours beaucoup moins d'énergie en mode veille. BattLab-One vous donne la possibilité de commuter la plage de courant en mode veille entre 10 uA et 800 µA ou entre 800 µA et 500 mA pour une résolution supérieure de la mesure. Pour information, l'appareil ne doit pas consommer trop de courant dans la plage de mesure inférieure. Selon la documentation, un courant de charge dépassant 250 mA peut endommager le matériel. Lorsque le dispositif testé est en mode veille, nous pouvons appuyer sur *Capture* pour enregistrer le courant.

Les résultats obtenus après un test avec *Hygrosage* sont présentés dans la **figure 7**. À l'étape 4, le système affiche les valeurs calculées. À l'aide de ces données, nous pouvons effectuer des expérimentations avec les caractéristiques et la capacité de la batterie, ainsi qu'avec les temps d'activité et de veille. Il suffit ensuite de cliquer sur *Optimize* pour voir facilement à quel degré d'autonomie nous pouvons nous attendre pour le dispositif lorsqu'il sera déployé sur le terrain.

La prédiction de l'autonomie de la batterie sera raisonnablement précise pour les appareils qui fonctionnent avec une période d'activité et de sommeil définie. Certains appareils dépendent d'une interruption d'événement externe pour sortir du mode veille ; dans ce cas, nous ne pouvons calculer l'autonomie de la batterie qu'en utilisant une estimation de la fréquence à laquelle les interruptions peuvent se produire.

En résumé

Le matériel du BattLab-One est tout à fait utilisable. Il est cependant possible d'améliorer le logiciel. Puisqu'il est basé sur des éléments *open source*, on peut s'attendre à ce que son évolution bénéficie du soutien de la communauté. Cela vous donne également la possibilité d'explorer les rouages de BattLab, et même d'adapter et d'améliorer le logiciel si nécessaire pour qu'il réponde à vos propres besoins. En ce qui concerne le matériel, BattLab-One fonctionne bien pour l'essentiel, malgré quelques inconvénients mineurs. Son faible prix et ses performances élevées en font un bon complément au banc d'essais pour ceux qui développent des dispositifs IdO.

210473-04

Des questions, des commentaires ?

Envoyez un courriel à l'auteur (tamhan@tamoggemon.com) ou contactez Elektor (redaction@elektor.fr).

Contributeurs

Texte et illustrations : **Tam Hanna**
Rédaction : **Thomas Scherer, C. J. Abate**
Mise en page : **Giel Dols**
Traduction : **Pascal Godart**

PRODUITS

› BattLab-One – sourcemètre avec boîtier
www.elektor.fr/19757

LIENS

[1] BattLab-One : http://www.elektor.fr/battlab-one-battery-life-optimizer-with-enclosure
[2] Logiciel sur Github : http://github.com/petersdw1/BattLab-One.git
[3] Logiciel pour Windows 10 : https://github.com/petersdw1/BattLab-One/releases/tag/V1.2.3

traceur de fuites à la terre simple

Mesure de l'isolation du secteur

Elbert Jan van Veldhuizen (Pays-Bas)

Les disjoncteurs différentiels sont des dispositifs de sécurité essentiels de nos installations électriques domestiques, mais il arrive qu'ils se déclenchent sans raison apparente. Un testeur de résistance de terre (ou mégohmmètre) professionnel aide à localiser la fuite, mais cet article décrit un traceur de fuite de terre facile à construire, une alternative sûre, abordable et utile, utilisant des piles au lithium CR2032 standard pour la « haute tension ».

ATTENTION ! Travailler sur la tension du secteur peut être fatal. Le circuit décrit ici n'est pas destiné aux débutants. Vous ne devez pas le construire ni l'utiliser si vous n'avez pas l'habitude de travailler sur le secteur !

Les commotions électriques sont potentiellement mortelles et les disjoncteurs différentiels ont sauvé de nombreuses vies. Alors que nos aïeux connaissent des personnes électrocutées, par ex. à cause d'une douille de lampe cassée, il est peu probable que cela se produise avec une installation électrique moderne. Un disjoncteur différentiel (DD) détecte les différences de courant entre le fil sous tension *L* et le neutre *N* (c'est-à-dire les fuites vers la terre). Si cette différence est supérieure à 30 mA, le disjoncteur différentiel se déclenche. Un courant alternatif d'environ 30 mA traversant le corps humain peut suffire à provoquer un arrêt cardiaque ou de graves lésions s'il persiste plus d'une fraction de seconde. Les DD sont conçus pour déconnecter les câbles alimentés assez vite afin de prévenir les blessures graves [1].
Un DD ne se déclenche pas seulement quand une personne touche le fil sous tension. L'humidité peut également induire des fuites, par ex. en pénétrant dans une prise murale ou une lampe extérieure. La source d'un tel problème est souvent difficile à localiser. Une méthode « simple » consiste à localiser le problème par tâtonnement : couper le courant, débrancher les différents fils dans des boîtes de dérivation, remettre le courant et voir si le disjoncteur différentiel se déclenche toujours. Mais cette méthode a ses inconvénients. Il faut en effet couper et rétablir le courant à chaque fois que des fils sont (dé)connectés : le risque d'erreur et d'intervention accidentelle sur un circuit sous tension augmente. En outre, les fuites à la terre peuvent être intermittentes et les défauts difficiles à reproduire.

Pourquoi concevoir ce circuit ?

Le concepteur de ce projet avait des soucis avec le circuit électrique de la véranda de la maison de ses parents : le DD de ce circuit se déclenchait parfois sans raison apparente. Lorsqu'il est venu vérifier le circuit pendant un week-end, le DD est resté enclenché pendant le test, mais s'est de nouveau déclenché quelques jours plus tard. Au lieu de se contenter de voir si le disjoncteur différentiel se déclenche et à quel moment, mieux vaut mesurer le courant de fuite. Si on peut mesurer un courant de fuite inférieur à celui qui déclenche le DD, il est beaucoup plus facile de s'attaquer aux problèmes (potentiels).

Comment mesurer

En un mot, en cas de défaut d'isolation d'un circuit du secteur, la mesure de résistance avec un multimètre standard ne permet pas de conclure. Quiconque a mesuré la résistance de sa peau avec un multimètre connaît ce paradoxe : la résistance mesurée est de l'ordre du MΩ (selon le degré d'humidité ou de transpiration des mains). On devrait en conclure qu'en touchant le 230 V du secteur, le courant serait bien inférieur à 1 mA, donc loin d'être mortel ; une telle intensité n'est même pas perceptible. Toutefois, la résistance dépend fortement de la tension appliquée pour la mesurer. Avec la plupart des multimètres, cette tension ne dépasse pas 9 V. En revanche, à 230 V, la résistance du corps humain est de l'ordre du kΩ, ce qui entraîne des courants de quelques centaines de mA qui sont dangereux, voire mortels. Cette dépendance de la résistance à la tension est due, entre autres, à des réactions électrochimiques dans le corps humain.
Ainsi, pour effectuer une mesure précise du courant de fuite, nous ne devons pas mesurer à une faible tension (comme le fait un multimètre) mais idéalement à 230 V, la tension du secteur dans la plupart des pays européens. Pour l'auteur, le défi était de le faire à faible coût, mais aussi, et c'est encore plus important, de façon sûre. À cet égard, une tension plus faible peut être conseillée et, compte tenu du mode d'alimentation choisi, le coût est réduit : plus la tension est faible, moins il faut de piles pour alimenter ce traceur de fuite à la terre.

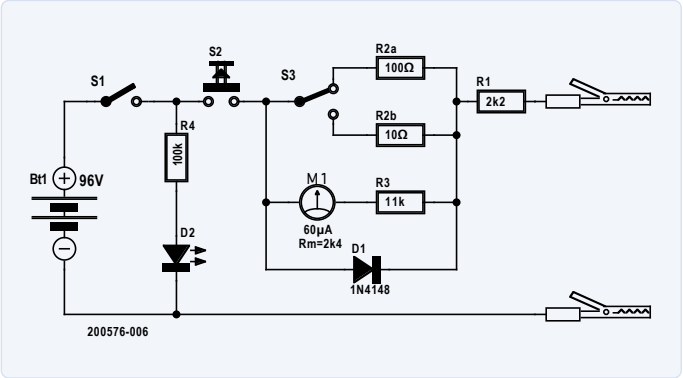


Figure 1. Schéma de principe du traceur de fuite à la terre.

Pourquoi des piles ?

La CR2032 est une pile au lithium de 3 V qui, comme son nom l'indique, fait 20 mm de Ø et 3,2 mm d'épaisseur. Le diamètre correspond à peu près à celui d'une pile C standard, qui mesure environ 50 mm de long. En empilant 16 CR2032 les unes sur les autres, on a à peu près le même facteur de forme qu'une pile C et une tension de 48 V. On peut donc utiliser un porte-piles standard pour deux piles C en série et obtenir 96 V. Les CR2032 sont largement utilisées dans les appareils électriques portables, les jouets et les gadgets : on les trouve presque partout à un prix très raisonnable. Pour un prix du même ordre, on pourrait utiliser 64 LR44, ou huit piles A23, mais il n'existe pas dans le commerce de porte-piles standard pouvant accueillir ces piles.
Au départ, un autre facteur en faveur des CR2032 dans cette réalisation était qu'elles ne peuvent délivrer qu'un courant assez faible, de l'ordre de 10 mA (selon certaines sources, mais il est difficile de trouver des données probantes à ce sujet !). 96 V_{cc} sont potentiellement dangereux, car en continu, le courant létal pour les humains est d'environ 100 mA. Selon les fiches techniques, les piles ne peuvent toutefois pas fournir un courant aussi élevé ; donc, en théorie, ces piles constituent une alimentation sûre pour le traceur de fuite à la terre. En pratique, l'auteur a cependant mesuré que des courants (de pointe) de quelques centaines de mA pouvaient être tenus assez longtemps pour être potentiellement dangereux (voir l'encadré sur les CR2032). C'est pourquoi cette réalisation inclut par sécurité une résistance de limitation de courant.

Le matériel

Pour mesurer le courant de fuite, l'auteur a utilisé un ancien galvanomètre encastrable qu'il avait encore sous la main. Un galvanomètre analogique a l'avantage de réagir plus vite qu'un appareil numérique (standard) et on voit les courants de pointe à la mise sous tension. La résistance de ce galvanomètre (R_m) est de 2,4 kΩ et la pleine échelle (I_{PE}) de 60 µA. Des résistances permettent de le faire fonctionner dans la gamme correcte. Mieux vaut le protéger contre les surintensités : si une diode limite à 0,7 V la ddp aux bornes de la branche *compteur* + *R3*, i_{max} = 53 µA (v. ci-après).

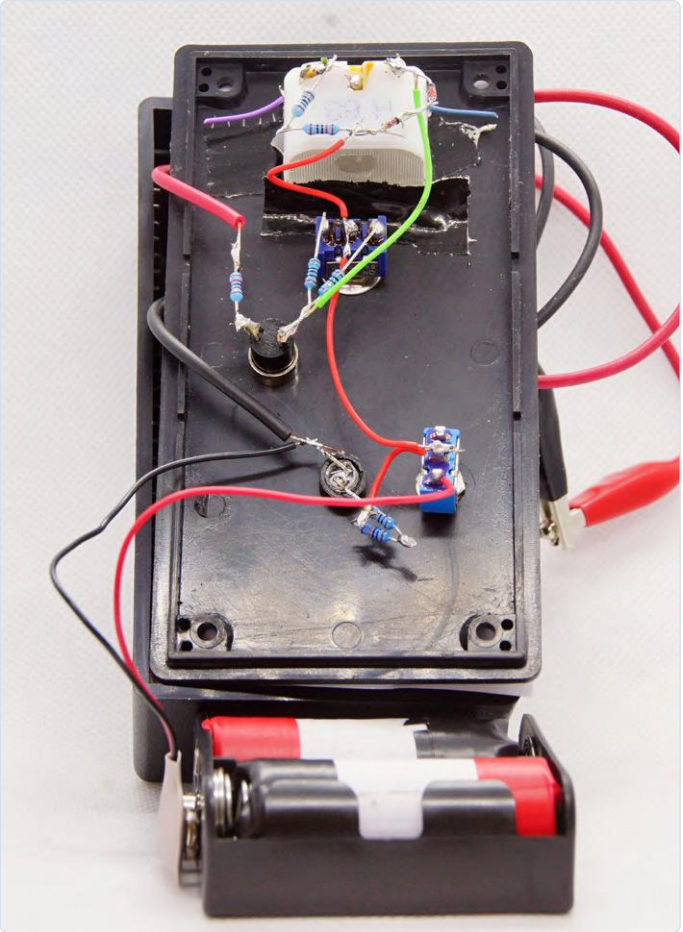


Figure 2. Réalisation du prototype de l'auteur.

COMBIEN DE CR2032 Y A-T-IL ?

Le succès de ce type de pile a un effet secondaire moins favorable : de nombreux fabricants produisent des CR2032. Outre les marques renommées, il existe un nombre inconnu de fabricants de piles au lithium de 3 V ayant les mêmes dimensions (20 mm de diamètre, 3,2 mm d'épaisseur), dont cette pile bouton tire son numéro de type. Si on mesure la capacité, la résistance interne et le courant maximal, on voit que les caractéristiques des CR2032 n'obéissent à aucune norme réelle. Et lorsqu'il s'agit des piles les moins chères (celles que vous préférez acheter quand il vous en faut 32), il est impossible de savoir de quel fabricant elles proviennent, et l'emballage ne fournit généralement pas plus de données électriques que « 3 V ». Rétrospectivement, les données générales trouvées par l'auteur indiquant que le courant de court-circuit de ces piles resterait de l'ordre de quelques dizaines de mA se sont avérées très optimistes du point de vue de la sécurité. En effet, les CR2032 qu'il a achetées avaient un courant de court-circuit d'environ 300 mA, qui a lentement chuté à 80 mA en dix secondes. Ce serait donc une mauvaise idée d'omettre la résistance R1 qui limite le courant du traceur de fuite à la terre. Mieux vaut prévenir que guérir !

La **figure 1** montre le schéma de ce traceur de fuite à la terre. R1 est la résistance de sécurité limitant le courant de fuite : avec 2,2 kΩ, le courant de fuite max. est de 45 mA environ. Si ce courant maximal circule, R1 dissipe 4 W. Donc soit on utilise une résistance de puissance, soit on coupe au bout d'un court instant.
Avec la résistance de shunt R2, la résistance R3 et la résistance interne R_m du compteur et à la condition $R2 << (R3 + R_m)$, on peut utiliser la loi d'Ohm pour calculer la gamme correcte du compteur :

$$R3 \leq \frac{V_{diode}}{I_{PE}} - R_m \qquad R2 = \frac{(R_m + R3) \times I_{PE}}{I_{échelle}}$$

où $I_{échelle}$ est le courant de sortie maximal réel du circuit.

Nous pouvons ensuite modifier les valeurs des résistances de manière à pouvoir utiliser des résistances standard (gamme E12). L'auteur a utilisé les composants suivants :

- R3 = 11 kΩ (10 kΩ + 1 kΩ en série). Cela limite le compteur à '7' sur une pleine échelle de '8' : la diode limite la tension aux bornes de *R3 + compteur* à 0,7 V, le courant max. vaut 53 µA environ, mais la valeur pleine échelle est de 60 µA. L'auteur utilise donc deux diodes en série soit une tension totale d'1,4 V, de façon à atteindre la pleine échelle, tout en limitant le courant à travers le galvanomètre à une valeur tolérable.
- R2a = 100 Ω (pour la gamme 8 mA)
- R2b = 10 Ω (pour la gamme 80 mA)

Étant donné la simplicité du circuit et le peu de composants, la réalisation du projet ne nécessite aucun circuit imprimé. La **figure 2** montre l'intérieur du traceur de fuite à la terre construit par l'auteur. En bas de cette photo, on voit les deux empilements de CR2032 enveloppés dans du ruban adhésif, dans un porte-piles standard pour deux piles C. Les bornes plus et moins de ces piles de 46 V maison sont repérées par du ruban adhésif respectivement rouge et noir.

Mesure du courant de fuite à la terre

L'utilisation de ce traceur de fuite à la terre est simple. En allumant l'appareil avec S1, la LED d'alimentation D2 s'illumine. En appuyant sur le bouton S2, on applique le potentiel de mesure aux câbles du traceur (mieux vaut utiliser des pinces crocodiles pour les connecter aux fils du circuit à tester). Le galvanomètre montre le courant de fuite éventuel. De cette façon, nous pouvons vérifier le courant de fuite de chaque fil des boîtes de jonction.
Avec les 96 V des batteries, le courant mesuré est bien sûr inférieur

au courant qui circulerait réellement dans la fuite à la terre pour une tension secteur de 230 V. En effet, en supposant que la résistance de la fuite reste pratiquement constante entre 96 et 230 V, un courant proportionnel à la tension circulera. Par ailleurs, R_{int} = (résistance interne de la batterie + shunt + résistance de sécurité) réduit également le courant. Cette résistance interne peut facilement être déterminée en mesurant le courant de court-circuit du traceur de fuite avec un multimètre. Pour le prototype de l'auteur avec le shunt de 10 Ω (R2b) sélectionné, ce courant était de 38 mA et donc une résistance interne de 2526 Ω. À partir du courant mesuré I_m, le courant de fuite réel $I_{réel}$ avec la tension du secteur peut être calculé avec :

$$I_{réel} = \frac{V_{secteur}}{\frac{V_{batt}}{I_m} - R_{int}}$$

Report dans un tableau :

I_m [mA]	$I_{réel}$ [mA]
0	0
1	2
2	5
3	8
4	11
5	14
6	17
7	20
8	24
10	31
20	100
30	>200

Quel était donc le problème de l'installation électrique de la véranda de mes parents ? Un câble relié à une lampe d'extérieur présentait une fuite d'isolement qui indiquait 4 mA pendant le dépannage avec ce traceur de fuite à la terre, ce qui donnerait en pratique environ 11 mA avec la tension du réseau appliquée. La fuite s'est probablement aggravée pendant la pluie. La meilleure façon de résoudre le problème était de remplacer le câble.

200576-04

LIEN

[1] **Disjoncteur différentiel** : https://fr.wikipedia.org/wiki/Protection_diff%C3%A9rentielle

Contributeurs
Conception et texte : Elbert Jan van Veldhuizen
Rédaction : Luc Lemmens
Illustrations : Elbert Jan van Veldhuizen, Patrick Wielders
Mise en page : Giel Dols
Traduction : Yves Georges

Des questions, des commentaires ?
Envoyez un courriel au rédacteur (luc.lemmens@elektor.com) ou contactez Elektor (redaction@elektor.fr).

PRODUITS

- **Testeur numérique de dispositif à courant résiduel (RCD), 2710 de PeakTech**
  www.elektor.fr/19318
- **Testeur numérique de boucles, avec calcul du courant de court-circuit présumé (PSC), 2715 de PeakTech**
  www.elektor.fr/19078

ATTENTION ! HAUTE TENSION !

Toute intervention sur un circuit ou système (potentiellement) connecté à la tension du secteur présente un risque pour la sécurité. Ni Elektor ni le concepteur de ce projet ne pourront être tenus responsables de dommages (sous quelque forme que ce soit) consécutifs à l'utilisation de ce traceur de fuite à la terre. Comme pour tout projet impliquant une connexion au secteur, nous vous mettons sincèrement en garde : si vous ne savez pas ce que vous faites, ne le faites pas !

PAUVRETÉ et électronique

Quand le courant économique passe mal

Priscilla Haring-Kuipers (Pays-Bas)

Qu'elle soit positive ou négative, l'influence qu'exerce la technologie sur nos sociétés est multiforme. Cet article montre qu'en matière d'électronique, une politique de conception conduite et appliquée à bon escient peut aider à réduire la pauvreté.

« Éradiquer la pauvreté sous toutes ses formes et partout dans le monde », voilà le premier des *Objectifs de développement durable* définis par les Nations Unies. L'électronique participe à cet effort en fournissant des technologies et des moyens de communication essentiels à l'amélioration des pratiques agricoles, au micro-financement, ou encore à l'accès aux services de santé [1].

La famille Chowdhury vit en Inde avec 30 $ par mois. Leur objet préféré est ce téléphone mobile. (Source : Zoriah Miller, Dollar Street 2015 [2], licence CC BY 4.0)

Vivre avec 38 $ par mois

La première cible des Nations Unies est l'extrême pauvreté, soit le fait de vivre avec moins de 1,25 $ par jour. Si vous souhaitez vous faire une idée concrète d'un tel revenu, le site Dollar Street dresse le portrait de familles subsistant avec quelques dizaines de dollars par mois [2]. Durant l'année 2020, après des décennies de recul de la pauvreté, la pandémie de Covid-19 a malheureusement fait retomber quelque 120 millions de personnes sous ce seuil d'extrême pauvreté. Parmi les autres objectifs des Nations Unies figurent la réduction de moitié des personnes vivant sous le seuil de pauvreté défini par leur pays, l'accès aux services de base et à la propriété foncière pour tous, ainsi que la réduction de l'exposition aux évènements extrêmes liés au changement climatique.

Une mine de problèmes

Un des pires liens qu'entretient l'électronique avec la pauvreté est l'extraction des matières premières. Au Congo, celle du cobalt se fait

dans des conditions très éprouvantes pour les travailleurs – souvent des enfants – et contre un salaire qui les force à vivre dans une pauvreté extrême. Partout autour du monde, les mines d'or et de cuivre offrent des salaires très bas et des conditions de travail insalubres : le boom économique de l'industrie minière ne profite pas à ses travailleurs. Certains grands fabricants (Samsung, BASF, Tesla…) commencent à militer en faveur d'une activité minière éthique, mais pour l'instant aucun n'a dépassé le stade des déclarations de bonnes intentions.

La famille Jacques vit en Haïti avec 39 $ par mois. Ils utilisent ces piments pour leur cuisine. (Source : Zoriah Miller, Dollar Street 2015 [2], licence CC BY 4.0)

À nous de décider

Bon nombre des décisions prises dans les pays à revenu élevé déterminent le sort économique des personnes vivant dans les pays à faible revenu. Quelle législation appliquer pour les matériaux de l'électronique ? Que faire de nos e-déchets ? La réparabilité doit-elle être notre priorité ? Si nous voulons que les pays pauvres puissent réparer, reconditionner puis revendre nos rebuts au sein de leur propre cycle économique, il faut que nos produits soient conçus à cet effet. Si nous ne le faisons pas, ces pays ne tireront du tri de nos e-déchets que maigres salaires et conditions de travail insalubres. Nos choix de conception ont aussi une portée économique et sociale.

La technologie à la rescousse

Certains sont persuadés que la technologie sortira le monde de la pauvreté : « La technologie moderne est l'un des moyens les plus efficaces de lutte contre la pauvreté. » [3] La révolution de la technologie mobile a permis aux personnes pauvres d'accéder aux services bancaires mobiles et à des informations sanitaires vitales, aux agriculteurs de connaître le temps à venir et de mieux fixer le prix de leurs récoltes. Les mobiles permettent aussi d'avertir la population de l'imminence d'une catastrophe naturelle, notamment les personnes vivant dans des endroits reculés. Les panneaux solaires fournissent quant à eux une énergie fiable et permettent de pomper et filtrer l'eau souterraine. En favorisant de même les initiatives politiques et l'accès à l'éducation, les technologies actuelles peuvent aider les populations à sortir de leur (extrême) pauvreté.

« La révolution de la technologie mobile a permis aux personnes pauvres d'accéder aux services bancaires mobiles.

210561-04 – VF : Hervé Moreau

LIENS

[1] Département des affaires économiques et sociales des Nations Unies, « Rapport 2020 sur les objectifs de développement durable » : https://www.un.org/sustainabledevelopment/fr/progress-report/
[2] Dollar Street : https://www.gapminder.org/dollar-street
[3] The Borgen Project, « 10 Technological Solutions to Poverty », 2014 : https://borgenproject.org/10-technological-solutions-poverty/

elektorstore

www.elektor.fr

e-choppe Elektor

des produits et des prix surprenants

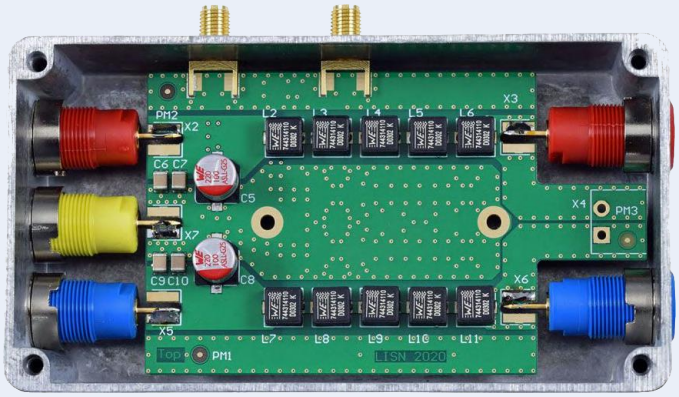
L'e-choppe Elektor a évolué. D'abord simple commerce de proximité pour nos productions (livres, magazines, kits et modules), c'est désormais une boutique en ligne bien rodée qui propose des produits surprenants à des prix très étudiés. Ce sont les produits que nous aimons et testons nous-mêmes. Si vous avez une suggestion, n'hésitez pas : sale@elektor.com. Seule exigence :
jamais cher, toujours surprenant !

Imprimante 3D résine Creality HALOT-SKY

Prix : 849,00 €
Prix (membres) : 764,10 €

www.elektor.fr/19745

Elektor Dual DC LISN 150 kHz – 200 MHz



Prix : 109,95 €
Prix (membres) : 98,96 €

www.elektor.fr/19869

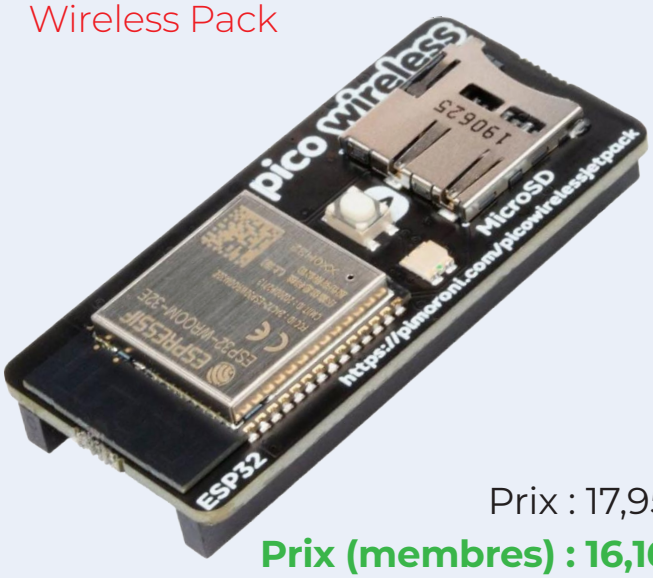
Cytron Maker Pi RP2040 – Robotique avec Raspberry Pi RP2040

Prix : 16,95 €

Prix (membres) : 15,26 €

www.elektor.fr/19926

Pimoroni Raspberry Pi Pico Wireless Pack



Prix : 17,95 €

Prix (membres) : 16,16 €

www.elektor.fr/19916

Kit de qualité de l'air MonkMakes pour Raspberry Pi

Prix : 29,95 €

Prix (membres) : 26,96 €

www.elektor.fr/19913

Carte d'évaluation WIZnet W5100S-EVB-Pico à base de RP2040

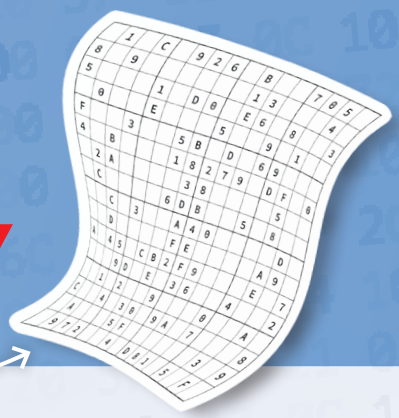
Prix : 14,95 €

Prix (membres) : 13,46 €

www.elektor.fr/19971

hexadoku

casse-tête pour elektorniciens



La dernière page de votre magazine propose toujours une grille de sudoku particulière. Éteignez le fer à souder, oubliez les écrans, rangez les pinces et les fils. Faites travailler votre matière grise d'une autre façon : attrapez un crayon pour remplir la grille d'hexadoku. N'oubliez pas d'envoyer votre réponse pour tenter de gagner un bon d'achat.

Une grille hexadoku est composée de chiffres du système hexadécimal, de 0 à F. Remplissez le diagramme de 16 x 16 cases de telle façon que **tous** les chiffres hexadécimaux de 0 à F (0 à 9 et A à F) n'apparaissent **qu'une seule et unique fois** dans chaque rangée, colonne et carré de 4 x 4 cases (délimités par un filet gras). Certains chiffres, déjà placés dans la grille, en définissent la situation de départ.

Pour participer, inutile de nous envoyer toute la grille, il suffit de nous donner **la série de chiffres** sur fond grisé.

Participez et gagnez !

Nous tirons au sort cinq des réponses internationales correctes reçues dans les délais ; leurs auteurs recevront chacun un bon d'achat Elektor d'une valeur de 50 €.

Où envoyer votre réponse ?

Envoyez votre réponse (les chiffres sur fond grisé) avec vos coordonnées par courriel, avant le **15 février 2022** à l'adresse **hexadoku@elektor.fr**

LES GAGNANTS

La solution de la grille du numéro de novembre/décembre 2021 est **53974.**

La liste des gagnants est publiée ici : www.elektormagazine.fr/hexadoku

Bravo à tous les participants et félicitations aux gagnants !

	4	7		3		B			5		0		6	8	
	1	D				0	2	B	7				A	4	
B	0		5				6	D				7		E	2
C	A		6		7	8	E	9	1	2		F		D	0
				1	8					5	9				
			A		3	7			C	B		8			
4		8	B				D	F				A	2		C
6	D	E											7	1	9
			3		A	F	9	C	4	0		D			
A						E			2						4
			2	8							A	C			
					5	6			9	D					
5	3	2	D	9	B	4			6	7	C	1	8	F	E
	B	A		0		5			3		E		D	C	
0			9	E							5	2			A
7	E	6		F	1	C			D	A	2		9	5	B

4	9	D	C	3	E	5	F	6	A	8	0	B	2	7	1
E	5	6	3	2	B	0	7	1	9	C	D	F	4	8	A
B	F	A	8	C	1	4	6	E	2	7	3	5	D	0	9
0	1	2	7	8	D	9	A	B	5	F	4	6	3	C	E
C	B	E	9	6	7	D	5	A	8	3	F	2	1	4	0
F	A	0	1	E	8	B	4	2	D	9	7	3	5	6	C
6	7	3	4	9	F	1	2	C	E	0	5	D	8	A	B
D	2	8	5	A	3	C	0	4	B	1	6	E	F	9	7
1	E	B	2	F	9	7	8	D	6	A	C	4	0	3	5
3	6	5	D	4	C	E	1	F	0	B	9	A	7	2	8
7	4	9	A	5	0	6	D	8	3	2	1	C	E	B	F
8	C	F	0	B	2	A	3	5	7	4	E	9	6	1	D
2	8	1	6	7	4	F	E	9	C	5	B	0	A	D	3
5	3	4	B	D	6	8	C	0	1	E	A	7	9	F	2
9	D	7	E	0	A	2	B	3	F	6	8	1	C	5	4
A	0	C	F	1	5	3	9	7	4	D	2	8	B	E	6

Tout recours est exclu, de même que le sont, de ce jeu, les personnels d'Elektor International Media et leur famille. Un seul gagnant par foyer.

elektor e-zine

Your dose of electronics

Chaque semaine où vous n'êtes pas abonné à l'e-zine d'Elektor est une semaine de grands articles et de projets électroniques qui vous manquent !

Alors, pourquoi attendre plus longtemps ? Abonnez-vous dès aujourd'hui à www.elektor.fr/ezine et recevez également le livre gratuit du projet Raspberry Pi !

À quoi pouvez-vous vous attendre ?

Éditorial
Chaque vendredi, vous recevrez les meilleurs articles et projets de la semaine. Nous couvrons les projets basés sur les MCU, l'IdO, la programmation, l'IA, et plus encore !

Promotionnel
Ne manquez pas les promotions de notre magasin, chaque mardi et jeudi nous avons une promotion spéciale pour vous.

Envoi des partenaires
Vous souhaitez rester informé des activités en cours dans le secteur ? Alors ce courriel vous donnera les meilleures informations. Non régulier, mais toujours le mercredi.

elektor
design > share > sell

Rejoignez les électroniciens de la communauté Elektor

Devenez membre elektor GOLD membership maintenant !

- accès à l'archive numérique depuis 1978 !
- 6x magazine imprimé Elektor
- 9x magazine numérique (PDF) dont Elektor Industry (EN)
- 10 % de remise dans l'e-choppe et des offres exclusives pourles membres
- le DVD annuel d'Elektor
- accès à plus de 1000 fichiers Gerber, collaboration avec les milliers d'électroniciens d'Elektor LAB, et une ligne directe avec nos experts !
- possibilité de voir votre projet publié ou vendu par notre boutique en ligne

Également disponible

abonnement elektor GREEN membership sans papier !

- accès à l'archive numérique d'Elektor
- 10 % de remise dans l'e-choppe
- 6x magazine Elektor (PDF)
- offres exclusives
- accès à plus de 1000 fichiers Gerber

www.elektormagazine.fr/membres

elektor
design › share › sell